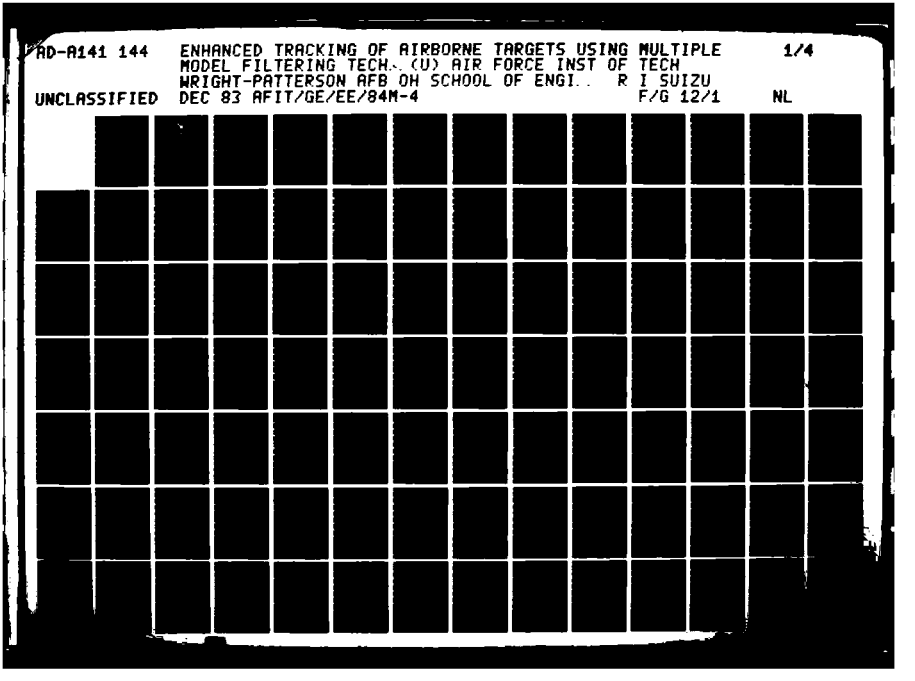
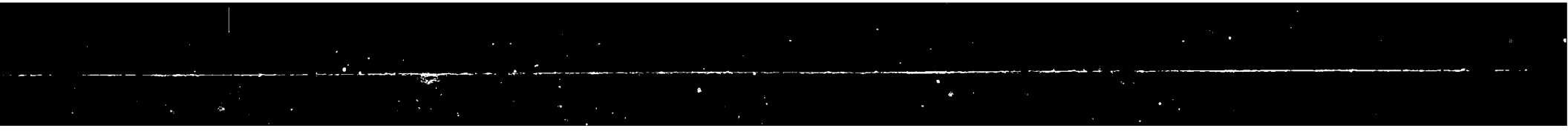
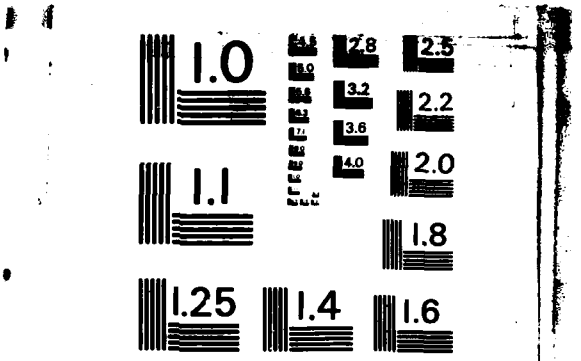
AD-A141 144 ENHANCED TRACKING OF AIRBORNE TARGETS USING MULTIPLE MODEL FILTERING TECH.. (U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. R I SUIZU 1/4

UNCLASSIFIED DEC 83 AFIT/GE/EE/84M-4 F/G 12/1 NL







MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

DTIC

AD-A141 144

ENHANCED TRACKING OF AIRBORNE TARGETS USING MULTIPLE MODEL FILTERING TECHNIQUES FOR ADAPTIVE FIELD OF VIEW EXPANSION

THESIS

Robert I. Suizu
First Lieutenant, USAF

AFIT/GE/EE/84M-4

DTIC FILE COPY

This document has been approved for public release and sale; its distribution is unlimited.

DTIC
ELECTE
MAY 17 1984
A

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

84 05 15 011

AFIT/GE/EE/84M-4

ENHANCED TRACKING OF AIRBORNE TARGETS USING MULTIPLE MODEL FILTERING TECHNIQUES FOR ADAPTIVE FIELD OF VIEW EXPANSION

THESIS

Robert I. Suizu
First Lieutenant, USAF

AFIT/GE/EE/84M-4

DTIC
ELECTE
MAY 1 7 1984

Approved for public release; distribution unlimited.

AFIT/GE/EE/ 84M-4

ENHANCED TRACKING OF AIRBORNE TARGETS USING MULTIPLE MODEL FILTER TECHNIQUES FOR ADAPTIVE FIELD OF VIEW EXPANSION

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Robert I. Suizu
First Lieutenant, USAF

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
DTIC COPY INSPECTED
A-1

December 1983

Approved for public release; distribution unlimited

Preface

This study was part of an ongoing effort at the Air Force Institute of Technology to design a tracking algorithm for use with the Air Force Weapons Laboratory's high energy laser weapon system. The purpose of this ~~study~~ thesis was to take previously developed tracker algorithms and incorporate a multiple model adaptive filter algorithm into the existing structure. This approach was intended to provide adaptive expansion of the effective tracker field of view, which in turn would increase the tracker's ability to maintain lock on highly dynamic, close range targets.

I wish to express my deepest thanks to my thesis advisor, Dr. Peter S. Maybeck, for his guidance, motivation, and above all, patience. His willingness to expend that extra effort to keep this research on track was invaluable to the completion of this study.

I would also like to thank my family, especially my parents, whose love and support keep me going when things aren't going well, and inspire me to do more when they are. I would also like to thank my fellow classmates in GE-83D, their friendship made the last 18 months at AFIT a bearable burden.

Table of Contents

Table of Contents

Table of Contents

List of Figures

List of Figures

List of Figures

List of Figures

List of Tables

List of Symbols

Symbol	
AR	aspect ratio
$\underline{A}$	direction cosine matrix
A_p	area of a single pixel
a,b	atmosphere break frequencies
$\underline{a}$	vector of uncertain parameter values
$\underline{a}_I$	inertial acceleration
α	smoothing constant in averaging process
$\alpha(t)$	azimuth
$\beta(t)$	elevation
$\underline{B}$	control input matrix
$\underline{C}_T$	truth model output matrix
E	sample mean error
E{ }	expected value
$\underline{e}$	unit vector
e	error between truth model system output and fil system output
$\underline{F}$	system plant matrix
$\underline{F}$[ ]	linearized function of the filter states
F( )	Fourier transform operation
F^{-1}( )	inverse Fourier transform operation
$\underline{f}$[ ]	nonlinear function of filter states
f_x, f_y	spatial frequencies
$\underline{G}$	system noise input matrix
$\tilde{G}$	complex-valued function in spatial frequency domain
$\tilde{g}$	complex-valued function in the space domain

List of Symbols

Symbol	
$\underline{h}[\ ]$	nonlinear intensity function
$\underline{H}[\ ]$	linearized intensity function
$\underline{H}$	system output matrix
$\underline{I}$	identity matrix
I_{max}	maximum intensity of the hot-spot
$\underline{K}$	Kalman filter gain
$\underline{\Phi}$	system state transition matrix
N x N	size of data array to be processed
ω	constant turn rate
ρ	range to target
$\underline{P}$	state covariance matrix or dispersion matrix of target image
p	conditional probability
$\underline{Q}_{FD}$	dynamics driving noise matrix
$\underline{Q}_{TD}$	noise covariance kernel descriptor for truth model dynamics
$\underline{R}$	measurement noise covariance matrix
$\underline{r}$	residual vector
r_h	horizontal range
$r_{k,l}$	correlation coefficient between k-th and i-th pixels
σ	standard deviation of a process or dispersion of an intensity distribution
T_{AF}	correlation time assumed for atmospheric jitter
T_{DF}	correlation time assumed for target dynamics
t,	time
Δt	sample time

List of Symbols

Symbol	
θ	orientation angle in the FLIR image plane
$\underline{u}$	deterministic control input
$\underline{v}$	measurement noise vector - zero-mean, white Gaussian noise
$\underline{v}_I$	inertial velocity vector
$\underline{v}'$	unit variance, zero mean, white, Gaussian noise vector
$\underline{w}$	white Gaussian noise vector
$\underline{x}$	general state vector
$\underline{x}_I, \underline{y}_I, \underline{z}_I$	inertial axes
$\Delta\underline{x}$	state vector update
ξ	angle between inertial velocity vector and the plane that is perpendicular to the line of sight
(x,y)	FLIR frame coordinates
$\underline{y}(t)$	current averaged data frame
$\underline{y}(t)$	current data frame
$\underline{Z}$	measurement process
$\underline{z}$	measurement vector
$\perp$	perpendicular
LOS	line of sight
rms	root mean squared
$\sqrt[C]{\ }$	Cholesky square root

Subscripts

A,a	atmospheric disturbances
AF	filter model for atmospheric jitter

List of Symbols

Symbol	
α	azimuth direction
β	elevation direction
C	centroid
D	target dynamics
DF	filter model for target dynamics
d	discrete time
F	filter
I	inertial reference frame
i	i-th time frame
k	k-th term or filter
kl	kl-th pixel
m	m-th hot-spot or ellipsoid
n	normalized
o	initial value
pv	direction perpendicular to the target velocity
ppv	direction mutually perpendicular to the target velocity and pv direction above
r	direction along line of sight to the target from the tracker
T	truth model
v	direction of target velocity vector

Superscripts

c	value after control application
^	estimate

List of Symbols

Symbol	
$-$	before measurement update
$+$	after measurement update
$\tilde{}$	complex variable

Abstract

Previous studies at the Air Force Institute of Technology have developed two tracker algorithms which provide significant improvements in tracker performance against close-range, highly-dynamic, airborne targets, over a currently used direct correlation method. Digital signal processing techniques are used to derive a target shape function from available sensor information. In one formulation, this shape function is used in the measurement update portion of an extended Kalman filter to determine the target position offsets from the center of the sensor field of view. In the other tracker, the offsets are derived and incorporated into the tracking algorithm by using the shape function as a template for an enhanced correlator/linear Kalman filter structure. Combining these offsets with any a priori target information allows the tracker to produce better target position estimates than achievable from a conventional correlator. This research investigates using a multiple model approach for the adaptive expansion of the effective tracker field of view as a means of increasing the dynamic range of the tracker. Two independent Kalman filters, each receiving measurement information from a shared sensor, generate target position estimates. The multiple models are created by tuning the respective filters for "best" performance at differing conditions of exhibited target behavior and differing the physical size of their respective fields of view. Adaptive expansion of the

tracker field of view is obtained by summing the weighted estimates of the two filters.

ENHANCED TRACKING OF AIRBORNE TARGETS USING MULTIPLE MODEL FILTERING TECHNIQUES FOR ADAPTIVE FIELD OF VIEW EXPANSION

I. Introduction

Even though lasers were first developed in the late 1950s, they have generally been considered as part of future technology by the public at large. Recent advances in laser technology have allowed their use in everyday life to increase steadily. Today, lasers are widely used in surgery, scientific laboratories, industry, and military applications, because of their ability to deliver almost instantaneously, finely focused, concentrated beams of energy onto a particular spot.

It is these charactristics that make the laser very attractive as a possible weapon system. Because of its almost instantaneous transmission of energy from the weapon to the target, it eliminates the need for computing the lead angle necessary for a ballistic projectile to intercept the target. However, a number of factors will affect a laser's effectiveness against any target. Some of these factors are associated with the laser itself, such as the strength or power of the beam. Others are related to the atmosphere, or the medium through which the beam must pass on its way to the target. These include diffusion, or spreading of the beam, as well as any other condition which alters or distorts the beam in any way. Finally, there is the target itself. Its composition and sensitivity to the energy de-

posited by the laser on the target will affect the laser's ability to neutralize it. These factors will determine the necessary power, concentration, and duration of the laser's radiation of a particular spot on the target needed to disable it effectively. Add to all the above factors any evasive maneuvers performed by the target, and the development of a ground-based, anti-aircraft/anti-missile laser weapon system becomes a task replete with obstacles.

Two significant obstacles to the development of an effective laser weapon system are precision pointing of the laser and accurate tracking of the target. It is not sufficient simply to "paint" the entire surface of the target with laser energy, nor is it currently possible to have a laser powerful enough to achieve the instantaneous, spectacular destruction of the target portrayed in fictional depictions of future warfare. As stated above, practical limitations of the laser, the distortion of the beam as it passes through the atmosphere, and the nature of the target itself, all make it necessary for the laser energy to be concentrated on a specific spot on the target for some finite amount of time before the target is disabled.

1.1 Background

The Air Force Weapons Laboratory at Kirtland AFB, New Mexico, is currently testing a high energy laser weapon system against airborne targets. These targets must be tracked despite the presence of several factors which can cause relative motion between the emitted beam and the

target. These include: target motion, mirror vibration, atmospheric jitter, and sensor measurement errors (15:2).

Target measurements are supplied by a Forward Looking Infra-Red sensor (FLIR). This sensor was chosen because it allows the tracker to gather target information passively, thereby preventing the target from detecting that he is being tracked.

The current research effort at the Weapons Laboratory uses a correlation tracker to process sensor information. At any one time, the tracker takes predetermined or previously gathered FLIR data and compares it with new information gathered at the current time. Cross correlation between these two sets of data is used to generate the relative position offsets from one frame of data to the next. These offsets are used to control gimbals which position the FLIR so that the target is centered within the sensor's field of view. Since the laser has been coupled to the FLIR through a shared aperture, this centering of the target within the FLIR's field of view also points the laser toward the target.

Though this correlation technique can be applied to a wide variety of targets because it relies only upon measured target data, it is highly susceptible to noise and takes no advantage of any a priori knowledge about the target. In most practical applications, however, information about the target can be supplied or at least estimated. If this information could be incorporated into the tracking algo-

rithm, performance should be improved. Furthermore, the effects of atmospheric disturbances on radiated waveforms is well-known and can be described statistically. This will allow separation of true target motion from apparent motion caused by the distortion of the infrared wave while it travels from the target to the sensor. This separation is important since the laser beam will not undergo the same distortion as it travels toward the target. Finally, statistical information about the FLIR measurement noise and background noise is available and can be used to improve the estimate of the target's position further.

In recent years, considerable effort has been spent at the Air Force Institute of Technology, to demonstrate the feasibility and accuracy of a tracker for this application that uses an extended Kalman filter. This algorithm has been tested in simulations against both long range targets (12) and short range air-to-air missile targets (2,3,5,12,13,15,17). This algorithm results in a significant improvement in tracking capability over the currently used correlation tracker.

Specifically, the study by Mercier (12) demonstrated the feasibility of using the extended Kalman filter in an algorithm against long range targets whose FLIR image plane intensity shape function could be modelled as an Airy disc (well approximated as a bivariate Gaussian) pattern due to a point source. Also due to these long ranges, the exhibited target dynamics would be very benign.

The next study, by Harnly and Jensen (3), concluded that while estimates of target position and atmospheric jitter were sufficient to track very benign target trajectories, additional estimates of target velocities and accelerations were needed to enable tracking of more maneuverable targets. Target intensity images were no longer portrayed as having circular equal intensity contours, but were now described as being elliptical. Additional modifications were made to the algorithm to allow adaptive estimation of the true sizes and shapes of ellipses.

Studies by Worsley (17) and Flynn (2) demonstrated that a constant turn rate target acceleration model was a more descriptive model of maneuvering airborne targets than either a Brownian motion or a first order Gauss-Markov target acceleration model. This model consistently produced less biased estimates than those produced by filters using the other acceleration models. However, the improvements in performance were not significant enough relative to the additional computational burden imposed by using the constant turn rate model, to warrant its use in all applications. Flynn's study (2) utilized the multiple model filter algorithm which was used in this research. The primary purpose of the research was to investigate if using a bank of independent Kalman filters (each tuned for optimum performance against a different target dynamics condition) and optimally combining the estimates produced by each filter, could produce a better estimate of target position than a

non-adaptive filter could. Combination of the state estimates is accomplished by weighting the estimates of each filter with a weighting factor based on the "correctness" of each filter dynamics model relative to true target dynamics. The major problem of Flynn's study was that, because of the similarity of the residuals from all the filters, the algorithm was consistently unable to identify the filter with the "best" model.

Studies by Singletery (15) and Rogers (14) implemented algorithms which made no claims as to prior knowledge about the size or shape of the target hot-spots. Digital signal processing techniques were used to take FLIR data and compute an estimated target shape. In addition, Rogers' thesis demonstrated the feasibility of this technique against multiple hot-spot targets performing very benign trajectories. Rogers also developed an alternative tracker form, one which used the estimated target shape as a template against which later measurement data could be compared. This comparison is performed by an enhanced correlation algorithm whose output is provided as "measurements" to a linear Kalman filter. These "measurements" are the offset distances from center of the template to the point of maximum correlation in terms of FLIR image plane coordinates. A linear Kalman filter, as opposed to the non-linear extended Kalman filter, could be used because the offset distances are linear functions of the chosen filter states.

Because both techniques demonstrated similar capability against multiple hot-spot targets, follow-on studies by Kozemchak (5) and Millner (13) further developed these configurations by testing them against more realistic target trajectories. These trajectories included benign constant velocity trajectories and more dynamic constant-g, pull-up manuevers. The targets were also allowed to perform rolling maneuvers to test the trackers against multiple hot-spot targets whose target intensity functions in the FLIR image plane were changing constantly.

While the trackers in both of the above studies performed well against the simulated target maneuvers, both trackers had difficulty maintaining lock on targets performing pull-up maneuvers in excess of 5 g's. It is for this reason that investigation of additional techniques for processing target information was performed.

1.2 Problem

The purpose of this research was to take the tracker formulations developed by Kozemchak (5) and Millner (13) and determine the feasibility of implementing a multiple model filter structure as a means of adaptively adjusting the aperture of the tracker field of view to permit the tracking of targets performing highly dynamic maneuvers. As with Flynn's thesis (2), the different filters were tuned to achieve optimal performance at different degrees of target dynamics.

The data processing algorithms for the tracker confic rations will now be described. The first model is t extended Kalman filter tracker used by Kozemchak (5). T second model is the linear Kalman filter/correlator track used by Millner (13).

1.2.1 Extended Kalman Filter Tracker. Figure I illustrates the algorithm for the extended Kalman filt tracker implemented by Kozemchak (5). In this configurati the center of the FLIR field of view is positioned at t filter predicted target centroid location due to targ dynamics over the most recent sample period. Each frame FLIR data is arranged into a 64-dimensional measureme vector $\underline{z}(t_i)$, which is the input to the extended Kalm filter in the lower path of the figure. The extended Kalm filter uses the nonlinear and the linearized intensity fur tions ($\underline{h}[\underline{\hat{x}}(t_i^-),t_i]$ and $\underline{H}[\underline{\hat{x}}(t_i^-),t_i]$ respectively) to co pute an updated estimate of the state variables, $\underline{\hat{x}}(t_i$ from the measurement vector via the equation:

$$\underline{\hat{x}}(t_i^+) = \underline{\hat{x}}(t_i^-) + \underline{K}(t_i)[\underline{z}(t_i) - \underline{h}[\underline{\hat{x}}(t_i^-),t_i)] \quad (1-$$

where $\underline{\hat{x}}(t_i^+)$ = state estimate vector after measuremer incorporation at time t_i

$\underline{\hat{x}}(t_i^-)$ = state estimate vector propagated from previous measurement update to time t_i

$\underline{K}(t_i)$ = Kalman filter gain

$\underline{z}(t_i)$ = measurement vector of average intensit over individual picture elements (pixe of the FLIR array; the assumed measurement model is:

$$\underline{z}(t_i) = \underline{h}[\underline{x}(t_i),t_i] + \underline{v}(t_i)$$

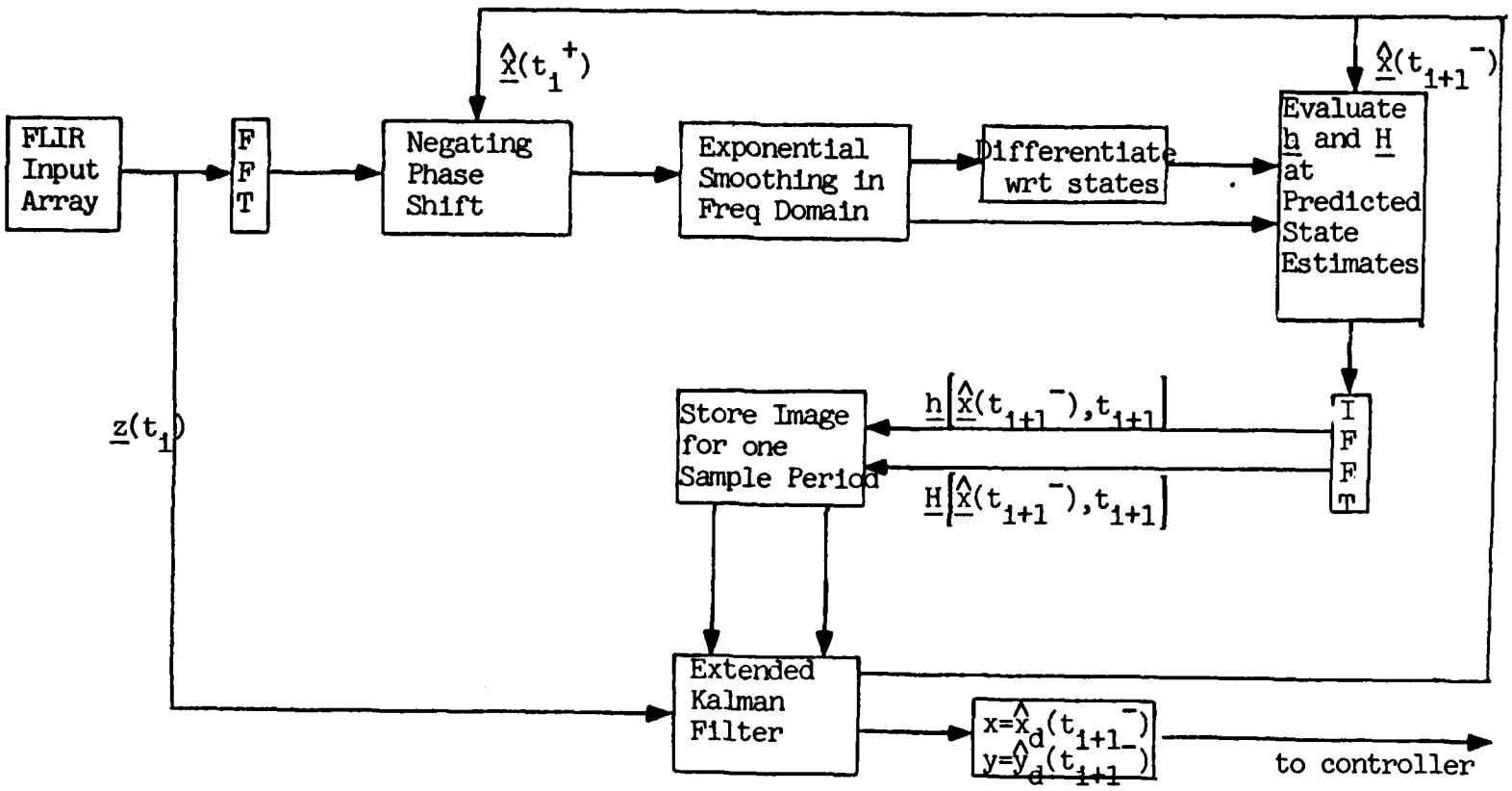


Figure I-1. Data Processing Algorithm for the Extended Kalman Filter Tracker

$\underline{h}[\underline{\hat{x}}(t_i^-), t_i]$ = intensity shape function for measurements at time t_i as a function of the state estimate

$\underline{H}[\underline{\hat{x}}(t_i^-), t_i]$ = $\partial \underline{h}/\partial \underline{x} \mid_{\underline{\hat{x}}(t_i^-)}$; used in generation of the Kalman filter gain $\underline{K}(t_i)$

$\underline{v}(t_i)$ = measurement corruption noise vector which includes FLIR measurement and background noises

The Kalman filter will then propagate this updated state estimate to the next sample time, t_{i+1}, based on its internal target dynamics model. This information is passed to the controller which positions the FLIR so that the center of its field of view is once again pointing to the predicted target position. For this research it was assumed that the controller was capable of pointing to the filter indicated position in the time available between samples (1/30 second).

Returning to the upper path of Figure I-1, a Fourier Transform is taken of the FLIR measurement data so that subsequent operations can be performed in the frequency domain. The motivation for carrying out those calculations in the frequency domain will be covered in a later chapter. Unlike the $\underline{z}(t_i)$ measurement vector formulated for the Kalman filter, the data is arranged in a 24 x 24 measurement array as opposed to an 8 x 8 array. This larger array was processed to reduce edge effects, aliasing, and leakage conditions involved when transforming finite sequences (15:18). In most engineering applications, the larger array is created by padding the original data with zeros. This is

a valid procedure as long as the image intensity is essentially zero at the edge of the 8 x 8 field of view. However, if this is not the case, padding with zeros will introduce artificial edge effects. In such cases, it is more appropriate to pad with data. Such padding is possible in this application because the field of view encompasses only a small portion of the entire FLIR measurement.

Generation of the nonlinear and linearized intensity functions requires interframe filtering to reduce the effects of noise. In order to perform this filtering, the target intensity profile must first be centered so that the noise can be averaged out. This centering of the target image is accomplished by multiplying the Fourier transform of the measurement data by a negating phase shift in the frequency domain. This negating phase shift is the complex conjugate of the linear phase shift corresponding to the (estimated) target image offset in the spatial domain. The causes of this offset are atmospheric jitter and imperfect propagation of the target dynamic states. This information is available from the updated state estimates of the extended Kalman filter.

Exponential smoothing is then performed on the centered data. Because the noise is expected to be changing far more rapidly than the target intensity pattern from one sample time to the next, this process will attenuate the noise by averaging the centered data of successive frames of data. The underlying target intensity function generated is used

as the reference image, $\underline{h}[\underline{\hat{x}}(t_i),t_i]$. The smoothed data is then differentiated with respect to a change in the Kalman filter states by employing the derivative property of the Fourier Transform to provide the frequency domain representation of the linearized intensity function, $\underline{H}$.

Because the nonlinear and linearized intensity functions will be used to update the Kalman filter estimates after the next measurement, both functions are evaluated at the state expected at the next sample time. This information is made available by propagating the updated Kalman filter states from the current measurement time to the next measurement time. Since it is assumed that the FLIR is centered on the position predicted due to estimated target dynamics, the intensity patterns are evaluated at the location of the predicted atmospheric states. Again, the shift theorem of the Fourier Transform is employed to perform the phase shift in the frequency domain. The inverse Fourier transform is then performed and $\underline{h}[\underline{\hat{x}}(t_{i+1}^-),t_{i+1}]$ and $\underline{H}[\underline{\hat{x}}(t_{i+1}^-,t_{i+1}]$ are ready for the extended Kalman filter to use when processing the next frame of data.

1.2.2 Linear Kalman Filter/Correlator Tracker. The Linear Kalman Filter/Correlator Tracker was the tracker initially developed by Rogers (14) and later extended and implemented by Millner (13). As can be seen in Figure I-2, it is very similar in structure to the extended Kalman filter tracker described in Section 1.2.1. The derivation of the target reference image is accomplished as before.

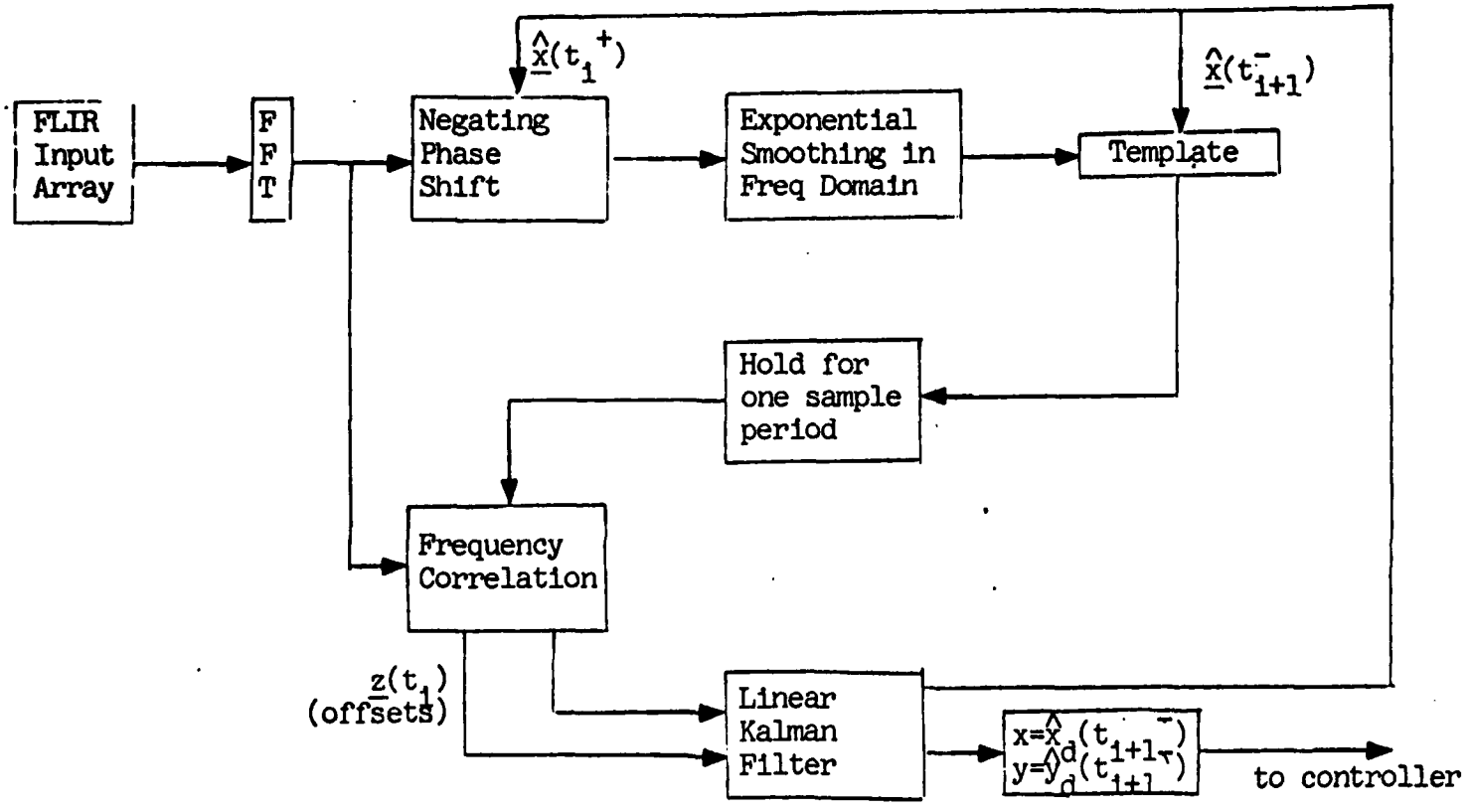


Figure I-2. Data Processing Algorithm for the Linear Kalman Filter/Correlator Tracker

The difference is that it is now used as the template of a correlator against which new data received from the sensor will be compared.

Correlation is performed in the frequency domain and the outputs of the correlator are the position offsets of the target centroid from the center of the reference image. Because these offsets are linear functions of the chosen filter states, a linear Kalman filter can be used in place of the extended Kalman filter operating on raw FLIR data, as in the previous section. This results in significant reduction in the number of operations required to process measurement information. In addition, because the "measurements" to the Kalman filter are only the position offsets, the measurement vector is now only a 2-dimensional vector.

Filter state propagation from one sample time to the next is accomplished as before, but now measurement updates for the linear Kalman filter are done using:

$$\underline{\hat{x}}(t_i^+) = \underline{\hat{x}}(t_i^-) + \underline{K}(t_i)[\underline{z}(t_i) - \underline{H}\underline{x}(t_i^-)] \qquad (1\text{-}2)$$

where $\underline{\hat{x}}(t_i^+)$ = state estimate vector after measurement incorporation at time t_i

$\underline{\hat{x}}(t_i^-)$ = state estimate vector propagated from previous measurement update to time t_i

$\underline{K}(t_i)$ = Kalman filter gain

$\underline{z}(t_i)$ = measurement vector of target centroid offsets from the center of the FLIR field of view, as generated by the correlator of Figure I-2

$\underline{H}$ = linear combination of the states which contribute to the respective measurements

Pointing of the sensor is accomplished as before under the assumption that the controller is capable of pointing to the filter indicated position within one sample time is as described in the previous section.

1.3 Overview

Chapters II, III, and IV describe in detail the mathematical models used in the computer simulation. More specifically, Chapter II presents the truth model, which was the environment from which measurements were taken. Chapter III describes the Kalman filters, extended and linear, that were used in the respective tracker configurations. Chapter IV discusses the multiple model filter algorithm; why it was chosen and how it was incorporated into the existing tracker structures.

Chapter V presents a performance analysis of the both tracker types, with the multiple model filter algorithm in place, against a variety of target trajectories. Chapter VI presents the conclusions and recommendations drawn from this research.

II. Truth Model

2.1 Introduction

The truth model is the representation of the real world implemented by the researcher. It may not be the most complete model available to him, but it should embody all important aspects of the problem and reproduce the real world environment with good fidelity. In this study, the following processes were included in the truth model: atmospheric jitter, target dynamics and shape effects, and background and FLIR noises. These processes are of importance to the tracking problem because they describe the motion of the true target in inertial space, and the distortion of the target intensity function as it passes through the atmosphere, as well as any background and FLIR measurement noises which combine to produce the observed target image.

This chapter describes the truth model used in this study. It includes discussions on the model for atmospheric jitter, the target trajectories used, and the development of the target measurement model.

2.2 Target Centroid Offset Model

During target tracking, there are a number of effects which can create apparent motion between the target and the sensor. These include: target dynamics, sensor boresight error, FLIR system vibrations, and atmospheric jitter. For this study of a ground-based laser system, it was assumed that dominant modes of apparent target motion are those

associated with target dynamics and atmospheric disturbances. Therefore, a continuous-time target model which incorporates both target dynamics and atmospheric disturbances describes the apparent target motion.

The FLIR measurements are scalar quantities that represent the average intensity of the received image over each picture element (pixel). The tracking window used in this research consisted of an 8 x 8 array of pixels in the FLIR image plane. Although the measurements are passed to the extended Kalman filter as a 64-dimensional vector, target dynamics and atmospheric disturbances are described using an x-y coordinate frame in the two-dimensional FLIR image plane in units of pixels (where each pixel is 20 μrads by 20 μrads). Hence the x- and y- coordinates of the apparent target centroid in FLIR image plane coordinates are:

$$x_C = x_D + x_A \tag{2-1a}$$

$$y_C = y_D + y_A \tag{2-1b}$$

where x_C = x-coordinate of apparent target centroid

y_C = y-coordinate of apparent target centroid

x_D = x-offset due to target motion

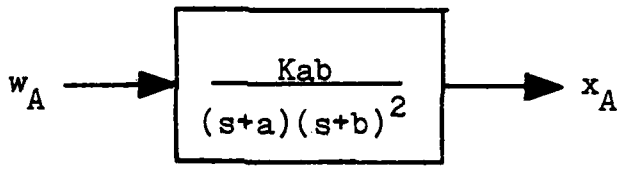
y_D = y-offset due to target motion

x_A = x-offset due to atmospheric disturbances

y_A = y-offset due to atmospheric disturbances

2.3 Atmospheric Disturbances

The target's intensity function will undergo translational motion on the FLIR image plane due to atmospheric disturbances which cause phase distortions in the radiated wavefronts from the target as they propagate through the atmosphere (14:27). The model used in this study was developed by The Analytical Sciences Corporation (16) and data analysis by Hogge and Butts (4). The power spectral density of this phenomenon in each of the two FLIR plane directions can be well approximated as the output of a third-order linear shaping filter driven by unit-strength, zero-mean, white Gaussian noise (12:12).



where w_A = unit-strength, zero-mean, white Gaussian noise

K = system gain

a = break frequency, 14.14 rad/sec

b = break frequency, 659.5 rad/sec

x_A = output of the shaping filter

By adjusting the value of the system gain, K, the desired root-mean-squared (rms) atmospheric characteristic can be obtained (12). The effects of atmospheric jitter are assumed independent of the direction on the FLIR image

plane, so two independent shaping filters of the above form can be used to model jitter; one for each coordinate direction of the FLIR image plane.

The developed mathematical model for atmospheric jitter will now be expressed in state space notation. Atmospheric jitter can be expressed in the time-invariant stochastic differential equation of the form:

$$\dot{\underline{x}}_a(t) = \underline{F}_a\underline{x}_a(t) + \underline{G}_a\underline{w}_a(t) \qquad (2\text{-}2)$$

where $\underline{F}_a$ = atmospheric plant matrix

$\underline{x}_a(t)$ = six atmospheric noise states

$\underline{G}_a$ = atmospheric noise input matrix

$\underline{w}_a(t)$ = two-dimensional vector of white Gaussian noise inputs with statistics:

$$E\{\underline{w}_a(t)\} = \underline{0}$$

$$E\{\underline{w}_a(t)\underline{w}_a^T(t+\tau)\} = \underline{I}\,\delta(\tau)$$

The shift of the intensity funciton due to atmospheric jitter can be expressed with:

$$\underline{x}_A(t) = \underline{H}_a\underline{x}_a(t) \qquad (2\text{-}3)$$

where $\underline{x}_A(t)$ = shift in FLIR coordinates, with components x_A and y_A as in (2-1)

$\underline{H}_a$ = system output matrix

Because of the independence between the disturbances occurring in the horizontal and vertical directions of the FLIR image plane, they can be decoupled and separate models can be developed. In Jordan canonical form, the disturbances in the x- (horizontal) direction becomes (12:73-75):

$$\dot{\underline{x}}_{ax}(t) = \begin{bmatrix} -a & 0 & 0 \\ 0 & -b & 1 \\ 0 & 0 & -b \end{bmatrix} \underline{x}_{ax}(t) + \begin{bmatrix} G1 \\ G2 \\ G3 \end{bmatrix} \underline{w}_{ax}(t) \qquad (2\text{-}4)$$

where a,b are the break frequencies described earlier and:

$$G1 = Kab^2/(a-b)^2$$

$$G2 = -G1$$

$$G3 = (a-b)G1$$

The output equation becomes:

$$x_{Ax}(t) = [\ 1 \quad 1 \quad 0\ ]\underline{x}_{ax}(t) \qquad (2\text{-}5)$$

The solution to the state differential equation (2-2) over one sample interval assumes the form:

$$\underline{x}_a(t_{i+1}) = \underline{\Phi}_a(t_{i+1},t_i)\underline{x}_a(t_i) + \int_{t_i}^{t_{i+1}} \underline{\Phi}_a(t_{i+1},\tau)\underline{G}_a(\tau)\underline{w}_a(\tau)d\tau \qquad (2\text{-}6)$$

where $\underline{\Phi}_a$ = the state transition matrix which is the solution to the matrix differential equation:

$$\dot{\underline{\Phi}}_a(t,t_i) = \underline{F}_a\underline{\Phi}_a(t,t_i)$$

and the initial condition:

$$\underline{\Phi}_a(t_i,t_i) = \underline{I} \text{ (the identity matrix)}$$

Since $\underline{F}_a$ is time-invariant, the state transition matrix is solely a function of the sampling time, $\Delta t = t_{i+1}-t_i$. Therefore, for a constant sampling time, Δt, the state-transition matrix is itself a constant. Thus, for the

system in (2-4) the state transition matrix for any time, t_i, is:

$$\underline{\Phi}_a(t_{i+1},t_i) = \begin{bmatrix} \exp(-a\Delta t) & 0 & 0 \\ 0 & \exp(-b\Delta t) & \Delta t\exp(-b\Delta t) \\ 0 & 0 & \exp(-b\Delta t) \end{bmatrix} \quad (2\text{-}7)$$

As before, the distortion in the y- (vertical) direction can be represented with a model of the same form.

Since it was desired to use a digital computer program to test the developed algorithms, an equivalent discrete-time system model of the continuous-time system was developed (6:42):

$$\underline{x}_a(t_{i+1}) = \underline{\Phi}_a(t_{i+1},t_i)\underline{x}_a(t_i) + \underline{w}_{ad}(t_i) \quad (2\text{-}8)$$

where $\underline{w}_{ad}(\cdot)$ is a discrete-time white Gaussian noise process with the identical statistics as the integral term of Equation (2-6) for all time:

$$E\{\underline{w}_{ad}(t_i)\} = \underline{0} \quad (2\text{-}9a)$$

$$E\{\underline{w}_{ad}(t_i)\underline{w}_{ad}^T(t_i)\} = \underline{Q}_{ad}(t_i)$$

$$= \int_{t_i}^{t_{i+1}} \underline{\Phi}_a(t_{i+1},\tau)\underline{G}_a\underline{I}\underline{G}_a^T\underline{\Phi}_a^T(t_{i+1},\tau)d\tau \quad (2\text{-}9b)$$

$$E\{\underline{w}_{ad}(t_i)\underline{w}_{ad}^T(t_j)\} = \underline{0} \text{ (for } t_i \neq t_j) \quad (2\text{-}9c)$$

Therefore the equivalent discrete-time system model can be expressed as (12:15):

$$\underline{x}_a(t_{i+1}) = \underline{\Phi}_a(t_{i+1},t_i)\underline{x}_a(t_i) + \sqrt[C]{\underline{Q}_{ad}}\ \underline{w}_n(t_i) \quad (2\text{-}10)$$

where $\sqrt[\mathcal{C}]{\underline{Q}_{ad}}$ is the lower Cholesky square root of $\underline{Q}_{ad}$ (6:3
71) and:

$$E\{\underline{w}_n(t_i)\} = \underline{0} \qquad (2\text{-}1$$

$$E\{\underline{w}_n(t_i)\underline{w}_n^T(t_j)\} = \underline{I} \qquad (2\text{-}1$$

$$E\{\underline{w}_n(t_i)\underline{w}_n^T(t_j)\} = \underline{0} \text{ (for } t_i \neq t_j) \qquad (2\text{-}1$$

2.4 Target Dynamics Models

The model for target dynamics is a continuous-ti
deterministic model which describes a highly maneuveral
aircraft or missile. In order to test the algorithm in
realistic environment as possible, a number of maneuv
were generated (5:35):

(1) straight line propagation

(2) constant roll-rate maneuvers

(3) constant G, constant speed turns

These manuevers will be tested individually to det
mine the algorithm's ability to follow a target perform
highly dynamic maneuvers. Later these same maneuvers w
be performed sequentially and/or simultaneously by the t
get so the algorithm can be tested against a realis
dynamic target whose intensity function is constan
changing.

Two previous studies, by Kozemchak (5) and Mill
(13) respectively, assumed that the centroid of the tar
intensity pattern coincides with the center of gravity
the target. Furthermore, the center of gravity of
target was assumed to be on the roll axis of the target,

rolling maneuvers performed by a non-pitching aircraft have no effect on centroid dynamics. Conversely, a pull-up maneuver does affect centroid dynamics for multiple hot-spot targets (5:35,13:22).

As developed by Harnly and Jensen (3), the true target location in the two-dimensional FLIR image plane can be expressed in azimuth, $\alpha(t)$, and elevation, $\beta(t)$. This true location can be compared with the filter estimates to evaluate filter performance. As with Kozemchak (5) and Millner (13), the azimuth and elevation rate inputs are used for propagating the true position as well as determining the accuracy of the filter's acceleration estimate. Thus, the time history of the target location can be generated via:

$$\dot{\underline{x}}_D = \underline{u}(t) = \begin{bmatrix} \dot{\alpha}(t) \\ \dot{\beta}(t) \end{bmatrix} \qquad (2\text{-}12)$$

where $\dot{\alpha}(t)$ and $\dot{\beta}(t)$ are the time varying azimuth and elevation velocities in inertial space. While it would have been easier to input time histories for $\alpha(t)$ and $\beta(t)$ directly for this deterministic case, the above form was used so a stochastic model could be readily implemented if desired at a later time (13:15).

At this point Kozemchak (5) and Millner (13) differed in their respective approximations of the dynamic offset. Kozemchak used a second-order model which assumed constant acceleration over each sample period:

$$\underline{x}_D(t_{i+1}) = \underline{x}_D(t_i) + \dot{\underline{x}}_D(t_i)\Delta t + 0.5\ddot{\underline{x}}_D(t_i)\Delta t^2 \qquad (2\text{-}13)$$

Millner used a first order model which used the values of the azimuth and elevation velocities which correspond to the midpoint of the sampling interval:

$$\underline{x}_D(t_{i+1}) = \underline{x}_D(t_i) + \dot{\underline{x}}_D(t_i+\Delta t/2)\Delta t \qquad (2\text{-}14)$$

The approximations of the dynamic offset were kept intact for each of the respective filters. This means that the extended Kalman filter tracker and the linear Kalman filter tracker used different truth models. If the assumptions made in the derivation of each model are correct (i.e. constant acceleration over each sample period for the extended Kalman filter tracker), then there is very little difference between the two models for a small sample period. However, the Millner form is a more correct model because it uses only those terms that are independently specifiable. The acceleration term in (2-13) is derived from the change in elevation and azimuth velocities, $\dot{\alpha}(t)$ and $\dot{\beta}(t)$ respectively, over a given sample period.

The azimuth and elevation velocities are initially calculated in the inertial frame and must then be projected into the FLIR image plane. The relationship between these two frames of reference is shown in Figure II-1 (5:37).

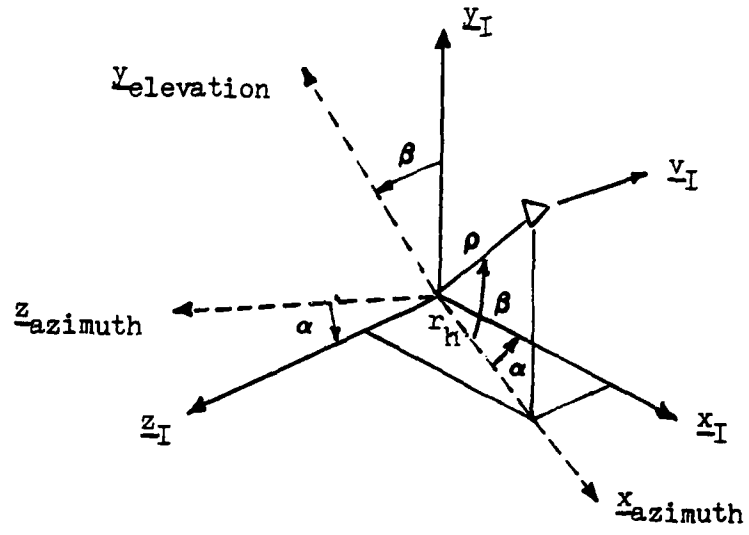


Figure II-1. Inertial Coordinate Frame

where $\underline{x}_I$, $\underline{y}_I$, $\underline{z}_I$ = inertial axes

ρ = range to target

$\underline{v}_I$ = target inertial velocity

r_h = horizontal range

α = azimuth angular displacement

β = elevation angular displacement

The geometry associated with azimuth direction is shown in Figure II-2 (13:19).

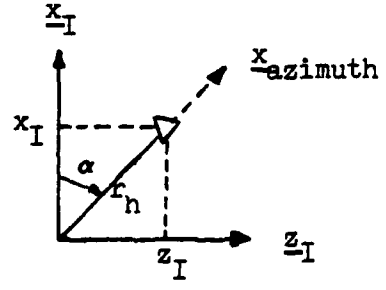


Figure II-2. Azimuth Geometry

Thus, the displacement angle, $\alpha(t)$, is defined as:

$$\alpha(t) = \tan^{-1}[z_I(t)/x_I(t)] \quad \text{(rad)} \tag{2-15}$$

and

$$\dot{\alpha}(t) = [x_I(t)\dot{z}_I(t) - z_I(t)\dot{x}_I(t)] \; / \; [z_I^2(t) + x_I^2(t)] \quad \text{(rad/sec)} \tag{2-16}$$

and

$$\ddot{\alpha}(t) = \{[x_I(t)\ddot{z}_I(t) - \ddot{x}_I(t)z_I(t)]r_h^2 - 2[x_I(t)\dot{x}_I(t) + z_I(t)\dot{z}_I(t)][x_I(t)\dot{z}_I(t) - \dot{x}_I(t)z_I(t)]\} \; / \; r_h^4 \quad \text{(rad/sec)} \tag{2-17}$$

where r_h is determined by $r_h = (x_I^2 + z_I^2)^{1/2}$. To convert these values into FLIR image plane units, pixels/sec, divide the number of radians by 20×10^{-6}. This conversion factor is derived from the fact that each pixel represents a region 20 μrads x 20 μrads (3:33).

Similarly, Figure II-3 displays the geometry involved in calculating elevation, elevation velocity, and elevation acceleration (13:20).

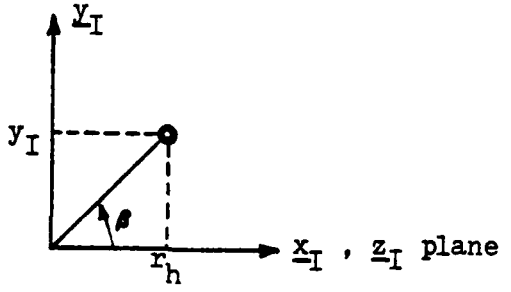


Figure II-3. Elevation Geometry

where $\rho = (x_I^2 + y_I^2 + z_I^2)^{1/2}$. Therefore,

$$\beta(t) = \tan^{-1}[y_I(t)/r_h(t)] \quad \text{(rad)} \tag{2-18}$$

and

$$\dot{\beta}(t) = [r_h(t)\dot{y}_I(t) - y_I(t)\dot{r}_h(t)] \,/\, \rho^2(t) \quad \text{(rad/sec)} \tag{2-19}$$

$$\text{where } \dot{r}_h(t) = [\dot{x}_I(t)x_I(t) + \dot{z}_I(t)z_I(t)] \,/\, r_h(t) \quad \text{(m/sec)} \tag{2-20}$$

and

$$\ddot{\beta}(t) = \{\rho^2(t)[r_h(t)\ddot{y}_I(t) - y_I(t)\ddot{r}_h(t)] - [r_h(t)\dot{y}_I(t) - y_I(t)\dot{r}_h(t)][2\dot{\rho}(t)\rho(t)]\} \,/\, \rho^4(t) \quad \text{(rad/sec)} \tag{2-21}$$

$$\text{where } \ddot{r}_h(t) = \{[x_I(t)\ddot{x}_I(t) + \dot{x}_I^2(t) + z_I(t)\ddot{z}_I(t) + \dot{z}_I^2(t)] \times r_h(t) - \dot{r}_h(t)[x_I(t)\dot{x}_I(t) + z_I(t)\dot{z}_I(t)]\} \,/\, r_h^2(t) \quad (\text{m/sec}^2) \tag{2-22}$$

$$\dot{\rho}(t) = [x_I(t)\dot{x}_I(t) + y_I(t)\dot{y}_I(t) + z_I(t)\dot{z}_I(t)] \,/\, \rho(t) \quad \text{(m/sec)} \tag{2-23}$$

The values for $\beta(t)$, $\dot{\beta}(t)$, and $\ddot{\beta}(t)$ can be converted to FLIR image plane units by the same conversion factor procedure used for the azimuth terms.

The solutions to the dynamics differential equation (2-12) have the form:

$$\underline{x}_D(t_{i+1}) = \underline{\Phi}_D(t_{i+1}, t_i)\underline{x}_D(t_i) + \int_{t_i}^{t_{i+1}} \underline{\Phi}_D(t_{i+1}, \tau)\underline{B}_D(\tau)\underline{u}_D(\tau)d\tau \tag{2-24}$$

where $\underline{\Phi}_D(t,\tau)$ = state transition matrix for vehicle dynamics

$\underline{B}_D(\tau)$ = control input matrix

$\underline{u}_D(\tau)$ = control function for the truth model as defined in equation (2-12)

For digital computer implementation, accelerations approximated as piecewise constant between sampling times result in a piecewise linear function for $\underline{u}_D(t)$. Therefore Equation (2-24) can be expressed in discrete-time form as:

$$\underline{x}_D(t_{i+1}) = \underline{\Phi}_D(t_{i+1},t_i)\underline{x}_D(t_i) + \underline{B}_d(t_i)\underline{u}_d(t_i) \quad (2\text{-}24)$$

In Kozemchak's thesis he defined $\underline{B}_d(t_i)\underline{u}_d(t_i)$ as:

$$\underline{B}_d(t_i)\underline{u}_d(t_i) = \begin{bmatrix} \Delta t & 0 & 0.5\Delta t^2 & 0 \\ 0 & \Delta t & 0 & 0.5\Delta t^2 \end{bmatrix} \begin{bmatrix} \dot{\alpha}(t_i) \\ \dot{\beta}(t_i) \\ \ddot{\alpha}(t_i) \\ \ddot{\beta}(t_i) \end{bmatrix} \quad (2\text{-}26)$$

where, of course, $\dot{\alpha}$ and $\ddot{\alpha}$ are as given above and are not independently specifiable, and similarly for $\dot{\beta}$ and $\ddot{\beta}$.

Millner defined the quantity somewhat differently, with (more appropriately) independently specifiable components only:

$$\underline{B}_d(t_i)\underline{u}_d(t_i) = \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} \alpha(t_i + \Delta t/2) \\ \beta(t_i + \Delta t/2) \end{bmatrix} \quad (2\text{-}27)$$

2.5 Overall State Space Model

Combining the truth models for target dynamics and atmospheric distortions yields a single eight state discrete-time model which consisting of two dynamics states and

six atmospheric states.

$$\underline{x}_T(t_{i+1}) = \underline{\Phi}_T(t_{i+1},t_i)\underline{x}_T(t_i) + \underline{B}_d\underline{u}_d(t_i) + \underline{G}_d\underline{w}_d(t_i) \qquad (2\text{-}28)$$

where $\underline{\Phi}_T(t_{i+1},t_i) = \underline{\Phi}_T(t_{i+1}-t_i) = \underline{\Phi}_T(\Delta t) =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \exp(-a\Delta t) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \exp(-b\Delta t) & \Delta t\exp(-b\Delta t) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \exp(-b\Delta t) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \exp(-a\Delta t) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \exp(-b\Delta t) & \Delta t\exp(-b\Delta t) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \exp(-b\Delta t) \end{bmatrix}$$

$$\underline{x}_D(t_i) = [x_D(t_i)\ y_D(t_i)\ x_{1A}(t_i)\ x_{2A}(t_i)\ x_{3A}(t_i)\ y_{1A}(t_i)\ y_{2A}(t_i)\ y_{3A}(t_i)]^T$$

For Kozemchak's study:

$$\underline{B}_d(t_i) = \begin{bmatrix} \Delta t & 0 & 0.5\Delta t^2 & 0 \\ 0 & \Delta t & 0 & 0.5\Delta t^2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{u}_d(t_i) = [\alpha(t_i) \quad \beta(t_i) \quad \alpha(t_i) \quad \beta(t_i)]^T$$

For Millner's study:

$$\underline{B}_d(t_i) = \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\underline{u}_d(t_i) = [\dot{\alpha}(t_i+\Delta t/2) \quad \dot{\beta}(t_i+\Delta t/2)]^T$$

$$\underline{G}_d = \overset{c}{\sqrt{\underline{Q}_{TD}}}$$

where $\underline{Q}_{TD} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & Q1 & Q2 & Q3 & 0 & 0 & 0 \\ 0 & 0 & Q2 & Q4 & Q5 & 0 & 0 & 0 \\ 0 & 0 & Q3 & Q5 & Q6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & Q1 & Q2 & Q3 \\ 0 & 0 & 0 & 0 & 0 & Q2 & Q4 & Q5 \\ 0 & 0 & 0 & 0 & 0 & Q3 & Q5 & Q6 \end{bmatrix}$

$$Q1 = [G1^2/2a]\ (1-\exp(-2a\Delta t)$$

$$Q2 = [G1^2/(a+b)]\ [(1-\exp(-(a+b)\Delta t)\ (-2b/(a+b) - (a-b)\Delta t\exp(-(a+b)\Delta t)]$$

$$Q3 = [(a-b)/(a+b)]G1^2(1-\exp(-(a+b)\Delta t)$$

$$Q4 = [G1^2/2b]\ \{[(1-\exp(-2b\Delta t)](1-(a-b)/b + (a-b)^2/2b) + (a-b)(2-(a-b)/b)\Delta t\exp(-2b\Delta t) - (a-b)^2\Delta t^2 \times \exp(-2b\Delta t)]$$

$$Q5 = [G1^2(a-b)/2b]\ [(1 - \exp(-2b\Delta t)(a-3b)/2b - (a-b) \times t\exp(-2b\Delta t)]$$

$$Q6 = [(a-b)^2G1^2/2b]\ (1-\exp(-2b\Delta t)$$

$$G1 = Kab/(a-b)^2$$

with statistics:

$$E\{\underline{w}_d(t_i)\} = \underline{0}$$

$$E\{\underline{w}_d(t_i)\underline{w}_d^T(t_i)\} = \underline{I}\delta_{ij}$$

$$\underline{Y}_T(t_i) = \underline{C}_T\underline{x}_T(t_i) \qquad (2\text{-}29)$$

where $\underline{Y}_T(t_i) = \begin{bmatrix} x_{centroid} \\ y_{centroid} \end{bmatrix}$

$$\underline{C}_T = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

2.6 Target Coordinate Frames

In order to describe the dynamics of various parts of the target relative to its center of mass, additional coordinate frames needed to be defined. This was necessary because a realistic image on the FLIR plane must be generated for instances when a multiple hot-spot target performs various maneuvers.

Target frame - This frame has its origin at the target center of mass. One of its axes is the inertial velocity vector of the target. Another axis is defined as being out the right side of the aircraft perpendicular to the velocity vector and in the plane of the radiating sources of the target. For example, for an aircraft with wing mounted engines, if the engines hang from pods, then the radiating sources are not in the same plane as the target center of mass. This model approximates many multi-engined aircraft. The third axis is defined by a vector perpendicular to the plane formed by the previous two vectors. This coordinate frame will be expressed with unit vectors $\underline{e}_v$, $\underline{e}_{pv}$, and $\underline{e}_{ppv}$, respectively.

α-β plane - This frame originates at the target center of gravity and is perpendicular to the true line of sight from the tracker (located at the origin of the inertial coordinate system) to the target. This plane can be shown to be defined by unit vectors $\underline{e}_\alpha$ and $\underline{e}_\beta$ which are misaligned from the inertial frame by the angles α and β which were defined in Figure II-1. The third basis vector is aligned

along the line of sight to the target.

2.7 Target Trajectories

As mentioned previously, a number of deterministic target trajectories which incorporate a number of maneuvers were developed to provide as realistic targets for the tracking algorithm as possible. The basic trajectories used in this study are those used by Millner (13).

Trajectory 1 - This trajectory is depicted in Figure II-4 and is a benign trajectory in which the target flies a constant-heading, straight-and-level course, either wings-level or performing a constant roll-rate maneuver.

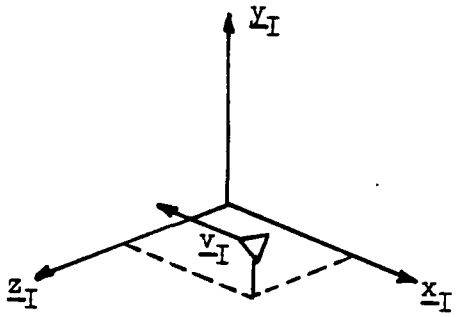


Figure II-4. Trajectory 1

The inertial velocity, $\underline{v}_I$, for this maneuver is held constant and is parallel to the x_I-z_I plane. Roll maneuvers are performed with clockwise rotation of the target as seen from behind.

Trajectory 2 - To evaluate filter performance against a more dynamic maneuver, a constant G pull-up maneuver was simulated. In this model the target begins with the same initial conditions as in the previous model. The maneuver is initiated at some pre-determined time after the simula-

tion begins to allow the tracker to obtain good target position estimates before the maneuver begins. This trajectory can be seen in Figure II-5.

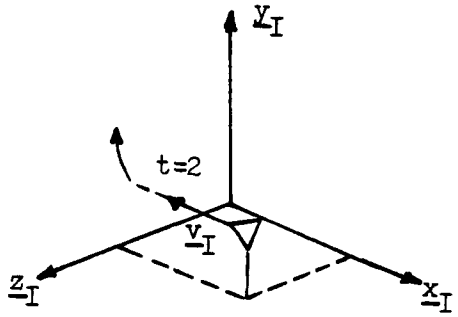


Figure II-5. Trajectory 2

It should be pointed out that this pull-up maneuver was started with a step change to the pitch-up rate, which is not a realistic model. However, this represents a more severe maneuver than any found in the real world, so the tracker should perform better against a more realistic target.

Trajectory 3 - This trajectory was used to evaluate performance with a target that begins and terminates a maneuver during a simulation. As with trajectory 2, a constant G pull-up maneuver is executed but instead of continuing the maneuver to the end of the simulation, it is terminated (again with a step change in the pitch-up rate) at some earlier time so the target returns to straight-line propagation. The target assumes whatever inertial velocity it possessed at the termination of the pull-up maneuver.

this velocity remains constant for the remainder of the simulation.

Trajectory 4 - This trajectory was used to provide a target which displays motion in all inertial directions. As was true for trajectory 3, this trajectory is similar to trajectory 2, but instead of terminating the maneuver as done in trajectory 3, the target turns toward the FLIR plane. This out-of-plane change maneuver causes the projected target image on the FLIR image plane to change its appearance more dramatically than in the previous cases, with substantial changes in the separation between the individual hot-spots.

2.7 Non-realistic Trajectories and Intensity Pattern Time Variations

Other areas of interest include non-realistic trajectories and intensity pattern variations. The desire here is not to portray realistic targets, but to determine the algorithm's sensitivity to various parameter changes. This can include investigation of performance against targets executing maneuvers beyond the capability of current aircraft, such as instantaneous and dramatic heading changes. Such a test could be useful in determining the tracker's ability to re-acquire a target that has shifted out of its field of view. Other parameter changes that can be investigated include: varying hot-spot size, intensity, separation between hot-spots, and variation of these parameters with time.

2.8 Measurement Model

The measurements provided to the tracker algorithm represent the intensity function generated by the target projected onto the FLIR image plane. This function is also corrupted by any background and FLIR noises that may be present. For distant targets, it was found that these target patterns could be modelled with a bivariate Gaussian function with circular constant-intensity contours (12). However, close range targets were found to be better modeled with similarly distributed contours of elliptical shape (3). As the target gets closer to the tracker positon, individual and separate hot-spots can be identified on the target, each modelled with the following intensity function:

$$I = I_{max}\exp\{-0.5[(x-x_{peak}) \quad (y-y_{peak})][\underline{P}]^{-1} \times [(x-x_{peak}) \quad (y-y_{peak})]^{T} \qquad (2\text{-}30)$$

where I_{max} = maximum intensity of the hot spot

x_{peak}, y_{peak} = coordinates of the peak intensity of the hot-spot

$\underline{P}$ = matrix whose eigenvalues are σ_v^2 and σ_{pv}^2, which corresponds to the dispersion of the elliptical contour in the target plane defined earlier, and whose eigenvectors dictate the angular orientation of the principal axes of these ellipses

The x- and y- coordinates for Equation (2-30) are calculated relative to the center of the tracker field of view.

For single hot-spot targets the centroid of the intensity function is assumed to coincide with the target center of mass. For multiple hot-spot targets the hot-spots are

distributed, for example, as shown in Figure II-6 (13:40)

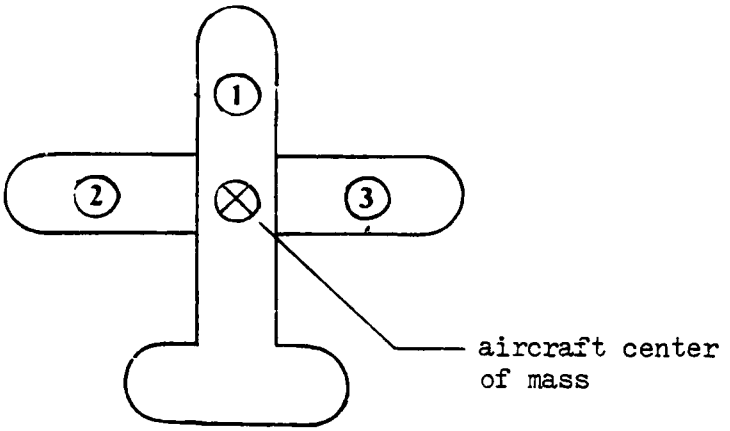


Figure II-6. Distribution of Hot-Spots

The multiple hot-spot case requires that the distance each hot-spot from the target center of mass be kno Also, it is assumed that the target side slip angle angle of attack are zero and that all the major axes of ellipsoids are parallel and aligned along the velocity v tor of the target, which extends out the nose of t vehicle.

For both the single and multiple hot-spot cases, measurements are generated by taking the average intens of each pixel in an 8 x 8 pixel tracking window, which due to the combined effects of each of the hot-spots and corruptive background and FLIR noises:

$$z_{kl}(t_i) = \sum_{m=1}^{M} [1/A_p \int_{kl\text{th pixel}} I_m(x,y,x_{peakm}(t_i),y_{peakm}(t_i))dxdy] + v_{kl}(t_i) \quad (2-$$

where $I_m(\cdot)$ = intensity function of the mth hot spot of M hot-spots

$z_{kl}(t_i)$ = output of the klth pixel at time t_i; the average intensity over that pixel as sensed by a detector in the FLIR image plane

A_p = area of one pixel

(x,y) = coordinates of any point within the klth pixel

$x_{peakm}(t_i)$, $y_{peakm}(t_i)$ = location of the peak of the mth intensity function at time t_i

$v_{kl}(t_i)$ = additive noise to the klth pixel corresponding to the background and FLIR noises

2.9 Target Image

It was assumed that the major axes of the m hot-spots are all aligned parallel to the inertial velocity vector. Furthermore, by assumption, all m hot-spots lie in the plane formed by the wings of the target.

As discussed previously, $\underline{v}_I$, the inertial velocity vector of the target is assumed to be projected out the nose of the target. The $\alpha-\beta$ plane is perpendicular to the true line of sight from the tracker to the target as defined by the basis vector $\underline{e}_r$. The origin of the target coordinate frame, the $\alpha-\beta$ plane, is the target center of mass. Figure II-7 illustrates the geometry involved.

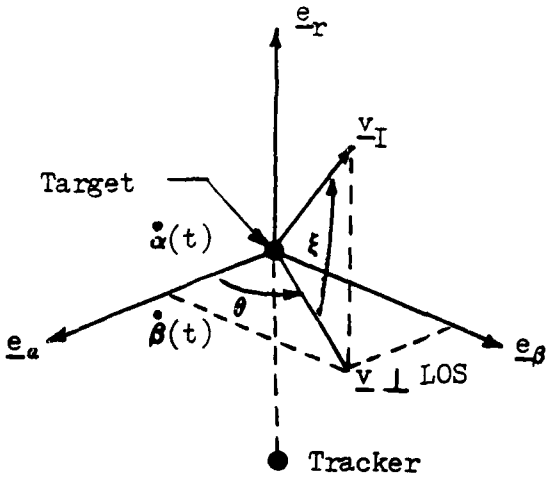


Figure II-7. Image Projection

From the figure it can be seen that

$$\cos\theta = \dot{\alpha}(t)/[\underline{v} \perp \text{LOS}] \qquad (2\text{-}32)$$

$$\sin\theta = \dot{\beta}(t)/[\underline{v} \perp \text{LOS}] \qquad (2\text{-}33)$$

where ($\underline{v} \perp$ LOS) is the magnitude of the velocity perpendicular to the tracker line of sight, i.e., the projection of $\underline{v}_I$ onto the α-β plane, defined by $[\underline{v} \perp \text{LOS}] = [\dot{\alpha}(t)^2 + \dot{\beta}(t)^2]^{1/2}$.

The size of the target and the distances between the hot-spots are fixed and do not vary with time. On the other hand, the image of the target as seen by the sensor will change as the target moves closer or farther away from the tracker or changes its orientation relative to the FLIR. This image at any time during the simulation is expressed in terms of some previously defined image, which has been specified at an initial range and size, and the current target range and velocity. The reference image is defined with the target lying flat in a plane perpendicular to the line of sight to the target. This produces an image with

the largest possible size. For example, if the reference image was flat and circular in shape, any other orientation will project an elliptical image of much smaller area onto the FLIR image plane. The following expressions relate the current image size with respect to the reference image:

$$\sigma_{pv} = \sigma_{pvo}\rho_o/\rho \tag{2-34}$$

$$\sigma_v = (\rho_o/\rho)[\sigma_{pvo} + \cos\xi\,(\sigma_{vo}-\sigma_{pv})]$$

$$= \sigma_{pv}\{1+[(\underline{v} \perp \text{LOS})/\underline{v}_I][AR-1]\} \tag{2-35}$$

where σ_{vo}, σ_{pvo} = the disperison of the target along the major and minor axes of the radiating ellipsoid, i.e., axes along and perpendicular to the velocity vector, respectively, for the reference image

σ_v, σ_{pv} = the current dispersions of the target image

ρ_o = reference range from sensor to the target

ρ = current range from sensor to the target

$\underline{v}_I$ = inertial velocity vector

$(\underline{v} \perp \text{LOS})$ = projection of $\underline{v}_I$ onto the α-β plane, the plane perpendicular to the line of sight to the target

ξ = angle between the inertial velocity vector, $\underline{v}_I$, and the α-β plane, as shown in Figure II-7

$AR = \sigma_{vo}/\sigma_{pvo}$ = maximum aspect ratio of the hot-spot reference image

As stated at the beginning of this section, the objective is to define the target image in terms of FLIR image plane coordinates. The relative distance between the hot-spots is known in the target frame defined in Section 2.6.

The coordinates of the centers of the hot-spots must be converted to α-β plane coordinates for use in (2-30) and (2-31). The transformation from the target frame coordinates to α-β coordinates is accomplished via:

$$\underline{x}_{\alpha\beta} = \begin{bmatrix} x \\ y \end{bmatrix}_{\alpha\beta} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{\text{target frame}} = \underline{A}\,\underline{x} \tag{2-36}$$

The dispersion matrix is transformed using:

$$\underline{P}_{\alpha\beta} = \underline{A}\,\underline{P}\,\underline{A}^T \tag{2-37}$$

Since it is desired to have the inverse of $\underline{P}_{\alpha\beta}$ for use in (2-30), a more convenient yet equivalent transformation can be generated by inverting this expression and using the orthogonal nature of $\underline{A}$ to yield

$$\underline{P}^{-1} = \underline{A}(\underline{P}^{-1})\underline{A}^T \tag{2-38}$$

2.10 Spatially Correlated Background Noise

The noise term, $v_{kl}(t_i)$, in Equation (2-31) associated with real data was found by Harnly and Jensen (3:19) to contain spatial correlation of the background noise with a correlation distance of about 2 pixels; this was modelled by maintaining non-zero correlation between each pixel and its two closest neighbors symmetrically in all directions. The 64 measurements (8 x 8 pixel array tracking window) are arranged as a 64-dimensional vector and so the spatially correlated noise can be modeled with:

$$\underline{v}(t_i) = \overset{c}{\sqrt{\underline{R}}}\, \underline{v}'(t_i) \qquad (2\text{-}39)$$

where $\underline{v}'(t_i)$ = a 64-dimensioned vector of independent white Gaussian noise processes with statistics:

$$E\{\underline{v}'(t_i)\} = \underline{0}$$

$$E\{\underline{v}'(t_i)\underline{v}'^{T}(t_j)\} = \underline{I}\delta_{ij}$$

The resulting noise process, $\underline{v}(t_i)$, has strength, $E\{\underline{v}(t_i)\underline{v}^{T}(t_j)\} = \underline{R}\delta_{ij}$, where $\underline{R}$ is the 64 x 64 matrix which describes the spatial correlation between pixels and is discussed in detail in the studies by Harnly and Jensen (3) and Kozemchak (5).

In order to generate the spatially correlated noise, the pixel numbering scheme illustrated in Figure II-8 was adopted. This array, which corresponds to the 8 x 8 FLIR data array, was numbered from 1 to 64 starting from the upper left hand corner of the array and proceeding across the rows as done by Harnly and Jensen (3:19).

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Figure II-8. Pixel Numbering Scheme

Harnly and Jensen (3) determined that the correlation coefficient matrix derived from using the above numbering scheme is:

$$\begin{vmatrix} 1 & r_{1,2} & r_{1,3} & \cdots & r_{1,64} \\ r_{1,2} & 1 & r_{2,3} & \cdots & r_{2,64} \\ r_{1,3} & r_{2,3} & 1 & \cdots & r_{3,64} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ r_{1,64} & r_{2,64} & r_{3,64} & \cdots & 1 \end{vmatrix}$$

Harnly and Jensen (3) also determined that the correlation terms not associated with the first and second nearest neighbors of a given pixel, could be approximated as essentially zero. For example, pixel 28 in Figure II-8, would have non-zero correlation terms only for pixels 10, 11, 12, 13, 14, 18, 19, 20, 21, 22, 26, 27, 28, 29, 30, 34, 35, 36, 37, 38, 42, 43, 44, 45, and 46. The measurement noise covariance matrix is obtained by multiplying the derived correlation coefficient matrix by the variance of the background noise.

Harnly and Jenson (3) also found that the effect of time correlation of the background noises is negligible at the anticipated signal-to-noise ratios. At this point, all the necessary terms for the measurement equation (2-31) have been developed in full.

2.11 Summary

This chapter has presented a number of models which have been used to model the real world. The processes that were modeled include: atmospheric jitter, target dynamics and shape effects, and background and FLIR noises. Deterministic target trajectories were introduced to provide baseline and realistic tracking scenarios to test the tracker algorithm. Finally, a number of coordinate frames were defined to aid in generating the appropriate realizations of the target intensity function in the FLIR image plane, and the entire truth model simulation was established.

III. Tracker Configurations

3.1 Introduction

The two tracker configurations, which served as the foundation of this research, were developed in studies by Kozemchak (5) and Millner (13). These trackers are sufficiently different that each must be presented separately in this chapter. Both of these trackers were first developed by Rogers (14) for multiple hot-spot targets performing very benign maneuvers. The later trackers extended the previous work to include targets performing much more highly dynamic and realistic trajectories.

The first tracker to be presented in this chapter is the one developed by Kozemchak (5). As shown in Figure I-1, it uses an extended Kalman filter to provide estimates of target position, velocity, and acceleration, as well as estimates of atmospheric disturbances. The need to include the estimate of target acceleration with the position and velocity estimates in a tracking scenario against highly maneuverable targets was shown by Harnly and Jensen (3). In order to maintain a reasonable computational burden for filter operation while still having the best possible target model, two different target models were used. The first used a simple first-order Gauss-Markov acceleration model (3,5,12,13,14,15,17), while the second used a constant speed, constant turn-rate model for target acceleration (5,17). While non-linear, the latter model has been shown to provide a much better description of real world target

dynamics when operating at short ranges, as in the case for air-to-air combat (2,17). Development of the equations needed to propagate the filter estimates and to update these estimates with FLIR measurement data will be presented later in the chapter.

While the above algorithm has tracked simulated targets adequately, the non-linearity of the problem requires a large number of equations to be processed on-line in real-time. If some sort of linear relationship in estimating the target parameters could be established, then many of the quantities needed to propagate and update the target estimates could be pre-computed and the on-line computational burden could be reduced. This desire for reducing the computational loading led to the development of an alternative tracker design to the extended Kalman filter tracker.

This alternative tracker uses the estimated target shape as a template in a correlator to provide pseudo-measurements to a linear Kalman filter, as shown in Figure I-2. These pseudo-measurements allow the use of the linear Kalman filter because they are in reality offset distances from the center of the field of view in the FLIR image plane. These distances are linear functions of the chosen state variables described in Chapter II (13). This tracker uses a first-order Gauss-Markov target acceleration model because the linear measurement model motivates use of a linear target descriptor, in order to yield a linear Kalman filter as the optimal data processor. Development of the

necessary equations to propagate and update these target estimates as well as a description of the correlation algorithm used will also be presented in this chapter.

3.2 Extended Kalman Filter Tracker

As stated earlier, this tracker uses an extended Kalman filter to provide estimates of target dynamics and atmospheric disturbances. Implicit in this use of the extended Kalman filter is the assumption that the non-linear target intensity function can be adequately linearized about the estimated states by using a Taylor series truncated to first order. Due to the relatively high measurment rate used in this research, such an approximation is considered valid.

3.2.1 Gauss-Markov Target Acceleration Model. The first of the filter models represents target dynamics acceleration and atmospheric jitter position as stationary, first-order Gauss-Markov processes. Such processes can be generated as the output of a first-order lag driven by zero-mean white Gaussian noise (6). A third-order model for atmospheric jitter position was presented in Chapter II, but because the double pole appearing in that model is sufficiently far away from the single dominant pole, it has little effect on the low frequency characteristics of the jitter (5). The filter vector states can be defined as target position, velocity, and acceleration, and jitter position, in each direction of the FLIR image plane:

$$\underline{x}_F = [x_D \quad y_D \quad v_x \quad v_y \quad a_x \quad a_y \quad x_A \quad y_A]^T \qquad (3\text{-}1)$$

The relationship between the states are as follows:

$$\dot{x}_D = v_x \tag{3-2a}$$

$$\dot{y}_D = v_y \tag{3-2b}$$

$$\dot{v}_x = a_x \tag{3-2c}$$

$$\dot{v}_y = a_y \tag{3-2d}$$

$$\dot{a}_x = (-1/T_{DF})a_x + w_{Dx} \tag{3-2e}$$

$$\dot{a}_y = (-1/T_{DF})a_y + w_{Dy} \tag{3-2f}$$

$$\dot{x}_A = (-1/T_{AF})x_A + w_{Ax} \tag{3-2g}$$

$$\dot{y}_A = (-1/T_{AF})y_A + w_{Ay} \tag{3-2h}$$

where T_{DF} = correlation time for target acceleration

T_{AF} = correlation time for atmospheric jitter

w_{Dx}, w_{Dy}, w_{Ax}, w_{Ay} = white Gaussian noise processes of zero mean and strength depending on the effect being modelled.

Note that identical independent models are used to represent effects in x- and y- directions of the FLIR image plane.

From the above relationships a state vector differential equation can be written in standard form:

$$\dot{\underline{x}}_F(t) = \underline{F}_F(t)\underline{x}_F(t) + \underline{G}_F(t)\underline{w}_F(t) \tag{3-3}$$

where $\underline{F}_F(t)$ = the system plant matrix which is constant for this application and can be written as:

$$\underline{F}_F = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1/T_{DF} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1/T_{DF} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/T_{AF} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/T_{AF} \end{bmatrix} \quad (3\text{-}4)$$

$\underline{G}_F(t)$ = the system noise input matrix which also is constant for this application

$$\underline{G}_F = \begin{bmatrix} \underline{0} \\ (4 \times 4) \\ \hline \underline{I} \\ (4 \times 4) \end{bmatrix} \quad (3\text{-}5)$$

$\underline{w}_F(t)$ = noise vector containing mutually independent white Gaussian noise processes w_{Dx}, w_{Dy}, w_{Ax}, w_{Ay} with statistics:

$$E\{\underline{w}_F(t)\} = \underline{0}$$

$$E\{\underline{w}_F(t)\underline{w}_F^T(t+\ )\} = \underline{Q}_F\delta(\ )$$

$$\underline{Q}_F = \begin{bmatrix} 2\sigma_{DF}^2/T_{DF} & 0 & 0 & 0 \\ 0 & 2\sigma_{DF}^2/T_{DF} & 0 & 0 \\ 0 & 0 & 2\sigma_{AF}^2/T_{AF} & 0 \\ 0 & 0 & 0 & 2\sigma_{AF}^2/T_{AF} \end{bmatrix} \quad (3\text{-}6)$$

σ_{DF}^2 = assumed target dynamics noise variance (or rms value)

σ_{AF}^2 = assumed atmospheric jitter noise variance (or rms value)

3.2.2 State Propagation of the Gauss-Markov Model. Due to the linear nature of the filter state model equations above, the state estimates can be propagated using standard Kalman filter propagation equations.

$$\underline{\hat{x}}(t_{i+1}^-) = \underline{\Phi}_F(t_{i+1},t_i)\underline{\hat{x}}(t_i^+) \tag{3-7a}$$

$$\underline{P}(t_{i+1}^-) = \underline{\Phi}_F(t_{i+1},t_i)\underline{P}(t_i^+)\underline{\Phi}_F^T(t_{i+1},t_i) + \int_{t_i}^{t_{i+1}} \underline{\Phi}_F(t_{i+1},\tau)\underline{G}_F\underline{Q}_F\underline{G}_F^T\underline{\Phi}_F^T(t_{i+1},\tau)d\tau \tag{3-7b}$$

where $\underline{\Phi}_F(t_{i+1},t_i)$ = filter state transition matrix for propagating from time t_i to time t_{i+1}

$\underline{P}(t_i^+)$ = conditional state covariance matrix after measurement update at time t_i

$\underline{P}(t_{i+1}^-)$ = conditional state covariance matrix before measurement update at time t_{i+1}.

The filter state transition matrix $\underline{\Phi}(t_{i+1},t_i)$ must satisfy the following differential equation:

$$\underline{\Phi}_F(t,t_i) = \underline{F}_F\underline{\Phi}_F(t,t_i) \tag{3-8}$$

over the interval (t_i,t_{i+1}), given the initial condition $\underline{\Phi}_F(t_i,t_i) = \underline{I}$. Because of the time invariant $\underline{F}_F$ matrix, the state transition matrix, $\underline{\Phi}(t_{i+1},t_i)$, is only a function of the sampling time Δt (= $[t_{i+1} - t_i]$) and can be evaluated via Laplace domain techniques or via matrix exponentials as:

$$\underline{\Phi}(t_{i+1},t_i) = \begin{vmatrix} 1 & 0 & \Delta t & 0 & J1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & J1 & 0 & 0 \\ 0 & 0 & 1 & 0 & J2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & J2 & 0 & 0 \\ 0 & 0 & 0 & 0 & J3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & J3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & J4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & J4 \end{vmatrix} \tag{3-9}$$

where $J1 = T_{DF}[\Delta t - T_{DF}(1-\exp(-\Delta t/T_{DF}))]$

$$J2 = T_{DF}[1-\exp(-\Delta t/T_{DF})]$$

$$J3 = \exp(-\Delta t/T_{DF})$$

$$J4 = \exp(-\Delta t/T_{AF})$$

The solution to the integral in Equation (3-7) becomes:

$$\underline{Q}_{FD} = \begin{bmatrix} A1 & 0 & A2 & 0 & A3 & 0 & 0 & 0 \\ 0 & A1 & 0 & A2 & 0 & A3 & 0 & 0 \\ A2 & 0 & A4 & 0 & A5 & 0 & 0 & 0 \\ 0 & A2 & 0 & A4 & 0 & A5 & 0 & 0 \\ A3 & 0 & A5 & 0 & A6 & 0 & 0 & 0 \\ 0 & A3 & 0 & A5 & 0 & A6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & A7 \end{bmatrix} \quad (3-1$$

where $A1 = \sigma_{DF}^2[2T_{DF}\Delta t^3/3 - 2T_{DF}^2\Delta t^2 - 4T_{DF}^3\Delta t\exp(-\Delta t/T_{DF}$
$+ 2T_{DF}^3\Delta t - T_{DF}^4\exp(-2\Delta t/T_{DF}) + T_{DF}^4]$

$A2 = \sigma_{DF}^2[T_{DF}\Delta t^2 + 2T_{DF}^2\Delta t\exp(-\Delta t/T_{DF}) + T_{DF}^3$
$- 2T_{DF}^3\exp(-\Delta t/T_{DF}) - 2T_{DF}^2\Delta t$
$+ T_{DF}^3\exp(-2\Delta t/T_{DF}]$

$A3 = \sigma_{DF}^2[-2T_{DF}^3\Delta t\exp(-\Delta t/T_{DF}) + T_{DF}^2$
$- T_{DF}^2\exp(-2\Delta t/T_{DF}]$

$A4 = \sigma_{DF}^2[2T_{DF}\Delta t - 3T_{DF}^2 + 4T_{DF}^2\exp(-\Delta t/T_{DF})$
$- T_{DF}^2\exp(-2\Delta t/T_{DF})]$

$A5 = \sigma_{DF}^2[T_{DF} - 2T_{DF}\exp(-\Delta t/T_{DF}) + T_{DF}\exp(-2\Delta t/T_{DF})]$

$A6 = \sigma_{DF}^2[1-\exp(-2\Delta t/T_{DF})]$

$A7 = \sigma_{AF}^2[1-\exp(-2\Delta t/T_{AF})]$

3.2.3 Constant Turn-Rate Target Acceleration Mode

The constant turn rate model has been shown to model t dynamics of real world airborne targets at close ran better than the Gauss-Markov acceleration model (2,17). tradeoff of using this improved model is the introduction

non-linearities into the propagation equations. The state differential equation becomes:

$$\dot{\underline{x}}_F(t) = \underline{f}[\underline{x}_F(t)] + \underline{G}_F\underline{w}_F(t) = \begin{bmatrix} v_x \\ v_y \\ a_x \\ a_y \\ -\omega^2 v_x + w_{Dx}(t) \\ -\omega^2 v_y + w_{Dy}(t) \\ (-1/T_{AF})x_A + w_{Ax}(t) \\ (-1/T_{AF})y_A + w_{Ay}(t) \end{bmatrix} \tag{3-11}$$

where ω = magnitude of the target's turn rate on the FLIR image plane derived via the relationship,

$$\omega = |\underline{v} \times \underline{a}|/|\underline{v}|^2 = (v_x a_y - v_y a_x)/(v_x^2 + v_y^2)$$

All other variables are as defined in the development of the Gauss-Markov target acceleration model. With the statistics of the noise vector being:

$$\underline{Q}_F = \begin{bmatrix} \sigma_{DF}^2 & 0 & 0 & 0 \\ 0 & \sigma_{DF}^2 & 0 & 0 \\ 0 & 0 & 2\sigma_{AF}^2/T_{AF} & 0 \\ 0 & 0 & 0 & 2\sigma_{AF}^2/T_{AF} \end{bmatrix} \tag{3-12}$$

The filter estimates are propagated forward in time by integrating:

$$\dot{\underline{\hat{x}}}(t/t_i) = \underline{f}[\underline{\hat{x}}(t/t_i)] \tag{3-13}$$

$$\dot{\underline{P}}(t/t_i) = \underline{F}[t;\underline{\hat{x}}(t/t_i)]\underline{P}(t/t_i) + \underline{P}(t/t_i)\underline{F}^T[t;\underline{\hat{x}}(t/t_i)] + \underline{G}(t)\underline{Q}(t)\underline{G}^T(t) \tag{3-14}$$

where (t/t_i) means at time t, given measurements through time t_i, starting from the initial conditions:

$$\underline{\hat{x}}(t_i/t_i) = \underline{\hat{x}}(t_i^+) \tag{3-15}$$

$$\underline{P}(t_i/t_i) = \underline{P}(t_i^+) \tag{3-16}$$

The non-linear function $\underline{f}[\underline{\hat{x}}(t/t_i)]$ can be shown to be equivalent to the time rate of change of the estimate of the state at time t. Furthermore, since there is a desire to keep the computational burden as low as possible, the integral of Equation (3-13) will be approximated with first order Euler integration, and so the state propagation equation becomes:

$$\underline{\hat{x}}(t_{i+1}^-) = \underline{\hat{x}}(t_i^+) + \dot{\underline{\hat{x}}}(t_i/t_i)\Delta t \tag{3-17}$$

The relationship assumes that the time rate of change of the state vector is piecewise constant during the time interval, Δt. This approximation is valid when the propagation time, Δt, is small compared to the natural transient times of the system.

To solve Equation (3-14) in real time would be both computationally burdensome and time consuming because of a time-variant $\underline{F}_F$ matrix which requires continuous re-calculation of the state transition matrix. For practical implementation, a first order Euler integration approximation was

made here as well. First, $\underline{F}[t;\underline{\hat{x}}(t/t_i)]$ is assumed to be piecewise constant during each time interval, Δt, and can be derived as a function of the current state estimate.

$$\underline{F}(t_i) = \partial \underline{f}[\underline{x}]/\partial \underline{x} \mid_{\underline{x} = \underline{\hat{x}}(ti^+)} \tag{3-18}$$

Given an invariant system plant matrix for each sample period, the state transition matrix can be determined. However, even this approximation requires significant numbers of operations to calculate $\underline{\Phi}_F(t_{i+1},t_i)$ and to determine the integral of Equation (3-14). To resolve this problem the upper-left 6 x 6 portion of the state transition matrix was truncated to first order terms.

$$\underline{\Phi}_F(t_{i+1},t_i) = \underline{I} + \underline{F}(t_i)\Delta t \tag{3-19}$$

The remaining portion of the state transition matrix is associated with the atmospheric jitter model and since it is time invariant, can be determined exactly, as J4 in Equation (3-9). The integral of Equation (3-15) can be similarly approximated with:

$$\underline{Q}_{FD} = \underline{G}_F \underline{Q}_F \underline{G}_F{}^T \Delta t \tag{3-20}$$

3.2.4 Measurement Update Equations. The measurement equation presented in Chapter II, (2-31), can be written in general form as:

$$z_{kl}(t_i) = h_{kl}[\underline{x}(t_i),t_i] + v_{kl}(t_i) \tag{3-21}$$

where $z_{kl}(t_i)$ represents the average intensity in the FLIR

image plane at the kl-th pixel at time t_i, h_{kl} represents the average intensity over that pixel due to the true target intensity, and v_{kl} represents the summed effects of background corruption and internal FLIR noise. The average intensities over each pixel in the FLIR image plane together form the target intensity shape function in terms of the most recent target measurement. The measurement information, provided at each of the 64 pixels in the tracking window, is used to update the filter and to produce a new estimate of the target centroid location in the FLIR image plane. The extended Kalman filter was used to incorporate the measurement information because of its ability to handle the nonlinearities of the problem and because it is less computationally burdensome than other non-linear filters (7:Ch 12).

The need to minimize computational loading also motivated the use of the inverse covariance form of the measurement update equations. Use of this form eliminated the need to perform a 64 x 64 matrix inversion at every update time (12). Thus, the update relations are:

$$\underline{P}^{-1}(t_i^+) = \underline{P}^{-1}(t_i^-) + \underline{H}^T(t_i)\underline{R}^{-1}(t_i)\underline{H}(t_i) \qquad (3\text{-}22)$$

$$\underline{P}(t_i^+) = [\underline{P}^{-1}(t_i^+)]^{-1} \qquad (3\text{-}23)$$

$$\underline{K}(t_i) = \underline{P}(t_i^+)\underline{H}^T(t_i)\underline{R}^{-1}(t_i) \qquad (3\text{-}24)$$

$$\underline{\hat{x}}(t_i^+) = \underline{\hat{x}}(t_i^-) + \underline{K}(t_i)\{\underline{z}(t_i) - \underline{h}[\underline{\hat{x}}(t_i^-), t_i]\} \qquad (3\text{-}25)$$

where $\underline{H}(t_i) = \partial \underline{h}[\underline{x}, t_i]/\partial \underline{x} \mid_{\underline{x} = \underline{\hat{x}}(ti^-)}$

the first partial of the average intensity function evaluated at the most

recent state estimate

$\underline{P}(t_i^-)$ = propagated conditional covariance matrix before measurement update at time t_i.

$\underline{P}(t_i^+)$ = conditional covariance matrix after measurement update at time t_i.

$\underline{K}(t_i)$ = Kalman filter gain

$\underline{\hat{x}}(t_i^-)$ = propagated state estimate before measurement update at time t_i.

$\underline{\hat{x}}(t_i^+)$ = state estimate after measurement update at time t_i.

$\underline{h}[\underline{\hat{x}}(t_i^-),t_i]$ = non-linear measurement function of average intensities at time t_i as a function of the most recent state estimate.

$\underline{z}(t_i)$ = actual realization of the measurement vector at time t_i.

The method used to derive the non-linear and linearized intensity functions will now be presented.

3.2.5 Derivation of Non-linear and Linearized Intensity Functions. The extended Kalman filter tracker uses the nonlinear intensity function $\underline{h}[\underline{\hat{x}}(t_i),t_i]$, and the linearized intensity function, $\underline{H}(t_i)$, to update the filter state estimates after each measurement, as shown in Equations (3-22) to (3-25). The method for deriving the nonlinear and linearized intensity functions will be outlined here.

All of the information of a two-dimensional intensity pattern can be represented by a set of eigenvalues and eigenfunctions. To obtain all the information contained in such a pattern may require an infinite number of such functions and values. Such a representation is unattractive because it cannot be practically implemented.

Ideally, it is desired to have a transformation which is not burdensome yet provides an accurate representation of patterns in an x-y coordinate system. In keeping with the truth models developed in Chapter II, it should also provide decoupling of the components in the new coordinate space.

The Karhunen-Loeve transformation is one such transformation. It generates a new coordinate space with perfectly uncorrelated elements. The major disadvantages of this technique are that it produces a correlation matrix of dimension $N^2 \times N^2$ for an N x N input matrix, and it is very difficult to perform in its exact form (14:15).

Such disadvantages encourage use of the Fourier transform. While this does not provide perfect decorrelation of the components, it is computationally attractive and possesses a property of separability which allows a two-dimensional transform to be obtained via one-dimensional operations (14:15).

3.2.6 Two-Dimensional Fourier Transform. In Section 1.2.1, it was explained that the target image had to be centered before averaging over successive frames of data could be performed to attenuate the noise. The derivative of the nonlinear intensity function was also needed to update the extended Kalman filter (see Figure I-1). The Fourier transform was used because it allows us to perform the centering (shifting) and derivative operations in the frequency domain where they are easily done. In the Fourier transform, complex exponentials are used as eigenfunctions

and the image is projected along the basis vectors associated with those exponentials.

The Fourier transform of a complex-valued function of two independent variables, $\tilde{g}(x,y)$, is a decomposition of $\tilde{g}(x,y)$ into a linear combination of functions of the form $\exp[j2\pi(f_x x + f_y y)]$ (15:8). The Fourier transform is defined by:

$$\tilde{G}(f_x,f_y) = F(\tilde{g}(x,y)) = \int_{-\infty}^{\infty}\int \tilde{g}(x,y)\exp[-j2\pi(f_x x + f_y y)]dxdy \quad (3\text{-}26)$$

where $\tilde{G}(f_x,f_y)$ = frequency spectrum, transformed function in spatial frequency domain

$\tilde{g}(x,y)$ = function in spatial domain

f_x,f_y = spatial frequencies

x,y = spatial variables

$F(\ )$ = Fourier Transform operation

The inverse of this transform also exists and is defined by:

$$\tilde{g}(x,y) = F^{-1}(\tilde{G}(f_x,f_y)) = \int_{-\infty}^{\infty}\int \tilde{G}(f_x,f_y)\exp[+j2\pi(f_x x + f_y y)]df_x df_y \quad (3\text{-}27)$$

where the terms are as defined above.

Because the FLIR provides target information as the average intensities over its exposed area, a two-dimensional discrete Fourier transform (DFT), is used. Due to of the separability of the Fourier transform, the two-dimensional transformation can be accomplished via a series of one-dimensional transformations, so the double integral can be resolved into a double summation for the discrete case. The

equations for the DFT and its inverse are:

$$H(f_x,f_y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} h(x,y)\exp[-j2\pi(f_x x + f_y y)] \qquad (3\text{-}28)$$

$$h(x,y) = 1/N^2 \sum_{f_x=0}^{N-1} \sum_{f_y=0}^{N-1} H(f_x,f_y)\exp[+j2\pi(f_x x + f_y y)] \qquad (3\text{-}29)$$

where $H(f_x,f_y)$ = frequency spectrum, transformed function in spatial frequency domain

$h(x,y)$ = function in spatial domain

f_x,f_y = spatial frequencies

x,y = spatial variables

The variable N refers to the period of the assumed recurring sequence in both directions. The assumption of a periodic sequence is essential in formulation of the DFT. Thus the complex sequence of intensity values is discretized into an N x N pixel array.

Although the tracking window is dimensioned to 8 x 8 pixels, the array that is processed by the DFT is dimensioned to 24 x 24. This is achieved by padding the data with a border of 8 zeros on each side of the tracking window. The purpose of this padding is to reduce the edge effects, aliasing, and leakage conditions involved when transforming finite sequences of numbers (15:18). Due to small area of the tracking window, it might not be appropriate to pad with zeros since the image intensity at the edges of the window might not be essentially zero. To pad such an image with zeros would introduce artificial edge

effects (5:10). However, since the tracking window is actually part of a much larger field of data from the FLIR, it is possible to pad the data of the tracking window with real data instead of zeros and thus minimize the introduced edge effects.

3.2.7 Shifting Property of the Fourier Transform. Since the target intensity pattern must be generated from noise corrupted FLIR data, interframe smoothing is necessary to attenuate the noise. This smoothing requires the target intensity profile to be centered from frame to frame since each pattern experiences different shifts from the center of the field of view. Successive centered frames of data can then be averaged to attenuate the noise, while at the same time accentuating the true target intensity function. Centering of each frame utilizes the shifting property of the Fourier transform as well as the filter's estimated location of the intensity profile.

The shift theorem for the Fourier transform states that a linear phase shift in the frequency domain corresponds to translation in the spatial domain. Because of the assumed periodic nature of the sampled data, such a phase shift can be thought of as a cylindrical shift. That is, rotation of the samples out one side of the interval results in rotating them into the other side of the interval. This property can be used to show that the only difference between the centered image and a translated image is a linear phase shift proportional to the spatial displacement in the x- and y-

directions as in (3-30), if $F(\tilde{g}(x,y)) = \tilde{G}(f_x,f_y)$, then

$$F(\tilde{g}(x-a,y-b)) = \tilde{G}(f_x,f_y)\exp[-j2\pi(f_x a + f_y b)] \quad (3\text{-}30)$$

where a = shift of the spatial function in the x-direction

b = shift of the spatial function in the y-direction

The filter's updated estimate of the location of the centroid of the intensity profile with respect to the center of the FLIR field of view can be used to determine the negating phase shift required to obtain the centered image needed for interframe smoothing.

3.2.8 Exponential Smoothing. The intensity profile of the target is neither known at any given time, nor can it be measured directly. Furthermore, the measurements that are available are corrupted by FLIR measurement noises as well as background noise. It is assumed that for most sampling rates, these corruptive noises tend to change significantly faster than the target intensity pattern from sample period to sample period (12).

A memory efficient, exponential smoothing algorithm was used to exploit this property. It captures the essence of a true finite memory averager without the need for storage of all the previous frames of data. This algorithm can be expressed as (1):

$$\underline{\hat{y}}(t) = \alpha \underline{y}(t) + (1-\alpha)\underline{\hat{y}}(t-1) \quad (3\text{-}31)$$

where $\underline{\hat{y}}(t)$ = current averaged value

$\underline{y}(t)$ = current data frame

$\underline{Y}(t-1)$ = previous averaged data frame

α = smoothing constant, $0 \leqslant \alpha < 1$

The smoothing constant, α, can be adjusted to account for the dynamics of the image. For slowly changing images, a smaller α would be appropriate, while a rapidly changing image requires that the most recent frames of data be weighted heavily. An α should be chosen that gives the best performance characteristics for all expected image variations. Appropriate values for α are: $0 \leqslant \alpha < 1$.

The necessary operations required to perform interframe smoothing have now been defined. A method for centering the intensity profiles and smoothing the data to attenuate the effects of noise has also been presented. The following operations are performed to get a centered intensity profile in the spatial domain.

1) The Fourier transform of the raw FLIR measurements is calculated

2) The appropriate negating phase shift is applied to center the image in the frequency domain based on the extended Kalman filter's estimate of the location of the centroid of the image

3) Interframe smoothing of the centered data is performed

4) The predicted target centroid position at the next sample time, $\underline{x}(t_{i+1}^-)$, is the sum of the predicted position due to target dynamics and the predicted position due to atmospheric disturbances. In equation form,

$$\underline{x}(t_{i+1}^-) = \underline{x}_{dyn}(t_{i+1}^-) + \underline{x}_{atm}(t_{i+1}^-)$$

However, as stated in Chapter I, control is applied at each sample time to zero the predicted dynamic states. That is, the FLIR is pointed so the center of the FLIR field of view points toward the

predicted target centroid position due to target dynamics. So now, the intensity shape function is evaluated at the states after control has been applied,

$$\underline{x}(t_{i+1}^{-\ c}) = \underline{x}_{atm}(t_{i+1}^{-})$$

where the superscript c denotes after controller application

5) The inverse Fourier transform is performed to obtain the intensity profile

3.2.9 Derivative Property of the Fourier Transform. In Section 3.2.4, the derivative of the intensity function with respect to the states was shown to be necessary to perform filter updates. This can be easily accomplished in the frequency domain, where differentiation in the spatial domain becomes simple multiplication by $j2\ (f_x + f_y)$. This process can be described by:

$$F[\partial \underline{h}(x,y)/\partial \underline{x}] = j2\pi f_x F[\underline{h}(x,y)] \qquad (3\text{-}32)$$

$$F[\partial \underline{h}(x,y)/\partial \underline{y}] = j2\pi f_y F[\underline{h}(x,y)] \qquad (3\text{-}33)$$

The necessary variables to propagate and update the estimates of the extended Kalman filter tracker have now been presented. The next sections of this chapter will cover the linear Kalman filter/correlator tracker configuration shown in Figure I-2, which shares many of the same processes described above.

3.3 Linear Kalman Filter/Correlator Tracker

As stated in the introduction to this chapter, a desire to obtain an algorithm that was less computationally burdensome to implement than a high-measurement-dimensioned

extended Kalman filter led to the investigation of the feasibility of using an enhanced correlator and Kalman filter in the same tracker. The correlator can be considered enhanced because it uses the estimated intensity function as a template against which to compare the new target information. Also, thresholding is performed on the cross-correlation to reduce the likelihood that false peaks will skew the estimate of the point of maximum correlation. This allows the correlator to incorporate a priori information about the target into the algorithm instead of operating solely on the collected data, as well as providing a better target template than the previous frame of raw FLIR data. Correlation of the template and the target information is used to estimate the relative position offsets from one sample period to the next. These pseudo-measurements are provided to the linear Kalman filter, which uses them to generate a new estimate of the target intensity function and target centroid location. Although Millner (13) investigated many different correlation techniques, the FFT method exhibited the best performance characteristics and was the method employed for this research.

3.3.1 FFT Correlator. The correlator used in this study computes the cross correlation of the template, which is the estimated target intensity function positioned at the best estimate of centroid offset ($\underline{h}[\underline{\hat{x}}(t_i^- - c]$), and the raw data from the FLIR. The FFT can be used to perform the cross-correlation as illustrated below:

$$F[\underline{g}(x,y)] = \underline{G}(f_x,f_y) \quad (3$$

$$F[\underline{l}(x,y)] = \underline{L}(f_x,f_y) \quad (3$$

$$F[\underline{g}(x,y)*\underline{l}(x,y)] = \underline{G}(f_x,f_y) \cdot \underline{L}^*(f_x,f_y) \quad (3$$

where $\underline{g}(x,y)*\underline{l}(x,y)$ = cross-correlation of the two dimensional spatial sequences $\underline{g}(x,y)$ and $\underline{l}(x,y)$

$\underline{L}^*(f_x,f_y)$ = complex conjugate of the Four transform of the sequence $\underline{l}(x$

By taking the inverse FFT, or IFFT, of Equation (3-36), cross-correlation is obtained:

$$\underline{R}(x,y) = \underline{g}(x,y)*\underline{l}(x,y) = F^{-1}[\underline{G}(f_x,f_y)\cdot\underline{L}^*(f_x,f_y)] \quad (3$$

Once the cross-correlation, $\underline{R}(x,y)$, has b determined, it may be necessary to process it wit threshholding function. If any one element of $\underline{R}(x,y)$ less than some pre-selected fraction of the element maximum correlation, then it will be considered as hav poor correlation information and be set to zero. T should reduce the likelihood of false peaks biasing estimated offset between the template and the target d While using a true maximum correlation finder wo eliminate this source of error, it is not attractive implement due to such problems as ambiguity of multi peaks and heavier computational loading, so a correlat peak was "found" by a center of mass correlation.

After threshholding, a centroid summation was use locate the center of mass of $\underline{R}(x,y)$. This centroid assumed to be a good indication of the peak location.

location of the autocorrelation centroid is calculated in either the x- or y- direction using the following equation:

$$C = \frac{\sum_{i=1}^{N} i \cdot amp_i}{\sum_{i=1}^{N} amp_i} \qquad (3\text{-}38)$$

The calculated position of the centroid of $\underline{R}(x,y)$ is the correlator's estimate of the offset of the target from the center of the data frame. This information is now the "measurement" passed to the Kalman filter. The appropriate measurement equation is:

$$\underline{z}(t_i) = \underline{H}_F \underline{x}_F(t_i) + \underline{v}_F(t_i) \qquad (3\text{-}39)$$

where $\underline{z}(t_i) = \begin{bmatrix} x_{DC} \\ y_{DC} \end{bmatrix} + \begin{bmatrix} x_{AC} \\ y_{AC} \end{bmatrix}$

= the estimate x- and y- coordinates of the centroid of the target intensity function as estimated by the correlation algorithm. The estimate of the centroid location is based on filter predicted centroid location due to dynamics and atmospherics.

$$\underline{H}_F = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

= the linear combination of the state variables which contribute to the measurement elements

$$\underline{x}_F(t_i) = [x_D \quad y_D \quad v_x \quad v_y \quad a_x \quad a_y \quad x_A \quad y_A]^T$$

$\underline{v}_F(t_i)$ = corruptive noise assumed to be a white Gaussian process with statistics

$$E\{\underline{v}_F(t_i)\} = \underline{0}$$

$$E\{\underline{v}_F(t_i)\underline{v}_F{}^T(t_j)\} = \underline{R}_F(t_i)\delta_{ij}$$

Recall that each measurement is considered to be the sum total of the position offsets from the center of the field of view due to target dynamics and atmospheric jitter, and the corruptive noise.

The appropriate propagation equations for this filter are the same as those developed earlier for the Gauss-Markov target acceleration model. This linear model was used because it is linear and, together with the linear measurement model, will yield a final filter that is totally linear. Because of the linearity of the equations in this tracker, standard Kalman filter update equations can be used:

$$\underline{K}(t_i) = \underline{P}_F(t_i^-)\underline{H}_F{}^T[\underline{H}_F\underline{P}_F(t_i^-)\underline{H}_F{}^T + \underline{R}_F(t_i)]^{-1} \quad (3\text{-}40)$$

$$\underline{\hat{x}}_F(t_i^+) = \underline{\hat{x}}_F(t_i^-) + \underline{K}(t_i)[\underline{z}(t_i) - \underline{H}_F\underline{\hat{x}}_F(t_i^-)] \quad (3\text{-}41)$$

$$\underline{P}_F(t_i^+) = \underline{P}_F(t_i^-) + \underline{K}(t_i^-)\underline{H}_F\underline{P}_F(t_i^-) \quad (3\text{-}42)$$

where all quantities have been defined previously.

3.4 Estimation of Dynamic Driving Noise

The dynamic driving noise matrix, $\underline{Q}_{FD}$, as in Equation (3-6), is used to model the expected effects of target motion. Because this motion is rarely constant, a constant $\underline{Q}_{FD}$ would be less than optimal most of the time for a target exhibiting a wide range of manuevers. Filter performance

would be improved if this matrix could be adaptively set in response to the observed target behavior. This adaptive estimation of the noise matrix can be employed for either tracker.

An estimator for $\underline{Q}_{FD}$ is determined as follows (71:Ch 10):

$$\underline{P}_F(t_i^-) = \underline{\Phi}_F(t_i,t_{i-1})\underline{P}_F(t_{i-1}^+)\underline{\Phi}_F^T(t_i,t_{i-1}) + \underline{Q}_{FD}(t_{i-1}) \tag{3-43}$$

which is a restatement of Equation (3-7b), substituting $\underline{Q}_{FD}(t_i)$ for the integral; t_i for t_{i+1}; and t_{i-1} for t_i.

$$\underline{P}(t_i^+) = \underline{P}_F(t_i^-) - \underline{K}(t_i)\underline{H}(t_i)\underline{P}_F(t_i^-) \tag{3-44}$$

Solving for $\underline{Q}_{FD}$ using Equations (3-43) and (3-44) yields:

$$\underline{Q}_{FD}(t_{i-1}) = \underline{K}(t_i)\underline{H}(t_i)\underline{P}_F(t_i^-) + \underline{P}_F(t_i^+) - \underline{\Phi}_F(t_i,t_{i-1})\underline{P}_F(t_{i-1}^+)\underline{\Phi}_F^T(t_i,t_{i-1}) \tag{3-45}$$

Only the first term of Equation (3-45) is not readily available because it is desired to have the Kalman filter gain reflect the observed target behavior. To incorporate this information into Equation (3-45) using the filter residual, $\underline{r}_F$, let

$$\underline{\hat{x}}_F(t_i^+) - \underline{\hat{x}}_F(t_i^-) = \underline{K}(t_i)\underline{r}_F(t_i) = \underline{\Delta x}(t_i) \tag{3-46}$$

If the ergodic assumption is made, the ensemble average can be replaced with a time averager on a single sample, as

$$E\{\Delta\underline{x}(t_i)\Delta\underline{x}^T(t_i)\} = 1/N \sum_{j=i-N+1}^{i} [\Delta\underline{x}(t_j)\Delta\underline{x}^T(t_j)] \quad (3\text{-}47)$$

Additionally, the residual sequence has been shown to be a white Gaussian sequence of zero mean and covariance $[\underline{H}(t_i)\underline{P}(t_i^-)\underline{H}^T(t_i) + \underline{R}(t_i)]$ (6:229).

$$E\{\underline{r}_F(t_i)\underline{r}_F^T(t_i)\} = \underline{H}(t_i)\underline{P}(t_i^-)\underline{H}^T(t_i) + \underline{R}(t_i) \quad (3\text{-}48)$$

so that

$$\begin{aligned} E\{\Delta\underline{x}(t_i)\Delta\underline{x}^T(t_i)\} &= \underline{K}(t_i)E\{\underline{r}_F(t_i)\underline{r}_F^T(t_i)\}\underline{K}^T(t_i) \\ &= \underline{K}(t_i)\underline{H}(t_i)\underline{P}(t_i^-) \end{aligned} \quad (3\text{-}49)$$

Combining Equations (3-45), (3-46), and (3-49) gives:

$$\begin{aligned} \hat{\underline{Q}}_{FD}(t_i) = \{1/N \sum_{j=i-N+1}^{i} [\Delta\underline{x}(t_j)\Delta\underline{x}^T(t_j)]\} + \underline{P}_F(t_i^+) \\ - \Phi_F(t_i,t_{i-1})\underline{P}_F(t_{i-1}^+)\Phi_F^T(t_i,t_{i-1}) \end{aligned} \quad (3\text{-}50)$$

But rather than averaging just for the first term of the above equation, averaging was performed over all terms for the N most recent sample periods:

$$\begin{aligned} \hat{\underline{Q}}_{FD}(t_i) = 1/N \sum_{j=i-N+1}^{i} [\Delta\underline{x}(t_j)\Delta\underline{x}^T(t_j) + \underline{P}_F(t_j^+) \\ - \underline{\Phi}_F(t_j,t_{j-1})\underline{P}_F(t_{j-1}^+)\underline{\Phi}_F^T(t_j,t_{j-1})] \end{aligned} \quad (3\text{-}51)$$

This is also a closed form approximation to the maximum likelihood estimate of $\underline{Q}_{FD}$ to be obtained simultaneously with a state estimate (7:123).

To reduce the data storage requirements, a fading memory approximation to this average was used:

$$\hat{\underline{Q}}_{FD}(t_i) = k\underline{Q}_{FD}(t_{i-1}) + (1-k)\underline{Q}_{FD1}(t_i) \qquad (3\text{-}52)$$

where $\hat{\underline{Q}}_{FD1}(t_i)$ = is a single term in the summation of Equation (3-51) when j=i

k = parameter which controls the length of time old estimates of $\underline{Q}_{FD}$ are maintained, $0 < k < 1$

3.5 Summary

This chapter presented the two tracker configurations which served as the foundations of this research effort. The developemnt of how state estimates were propagated and updated for both trackers was presented. Additionally, the use of the Fourier transform to derive the target intensity function was demonstrated.

The next chapter will discuss addition of a multiple model adaptive filter structure to the algoritms presented here.

IV. Multiple Model Adaptive Filter

4.1 Introduction

The studies by Kozemchak (5) and Millner (13) cited in earlier chapters used a tracking window much smaller than the entire area covered by each frame of FLIR data. Only an 8 x 8 array tracking window for measurement updates and a 24 x 24 array for data processing (out of an available 500 x 400 pixel measurement array) were used in an effort to minimize the computational and memory storage requirements of the tracker (3:4). Unfortunately, limiting the dimensions of the tracker field of view increases the likelihood that the image of a highly dynamic, close-range target will be outside the tracking window during a given sample period. Such a condition causes the tracker to lose track of the target. Any effort to increase the aperture of the single filter tracker without simultaneously increasing the array dimensions will decrease tracker resolution. A reduction of resolution results in poor tracker performance for benign target trajectories.

One approach which allows the tracker field of view to be expanded without increasing the dimensionality of the data processing arrays uses a multiple model filtering algorithm. Under this approach, a second identically dimensioned filter is processed in parallel with the original filter. The differences between the two filters are that the added filter possesses a different model of target dynamics and larger field of view. By optimally combining

the estimates produced by the two filters at each sample time, an overall estimate can be created. This structure will maintain the desirable high resolution for benign target trajectories while allowing the tracker to maintain lock on highly dynamic targets.

This chapter will describe the multiple model filter algorithm and how this algorithm was implemented in both the Kozemchak and Millner form of trackers.

4.2 Multiple Model Adaptive Filter Algorithm

In any tracking scenario there are many parameters related to target motion which are highly uncertain. Such parameters may be related to target capabilities and/or target commanded maneuvers. Furthermore, these uncertain parameters can be grouped together to form, $\underline{a}$; a vector which represents these uncertain parameters. The parameter vector $\underline{a}$ lies in a continuous parameter space, as each parameter composing $\underline{a}$ can generally assume any value over a continuous range of values. To make the estimation of $\underline{a}$ a more manageable task, this continuous space was discretized into K distinct models for the uncertain parameter vector. These models are well distributed over the range of expected values for $\underline{a}$ (7:130). The multiple model algorithm consists of K independent (Kalman) filters, each with its own estimate of the value of $\underline{a}$. That is, each filter's model for $\underline{a}$ contains values for each uncertain parameter, i.e. component of $\underline{a}$, that falls within the range of possible realizations of that parameter. Given that the filter whose $\underline{a}$ vector

most closely resembles the true values for the uncertain parameters should produce the best state estimates, one would, in a multiple filter environment, want to weight the estimates from that filter most heavily, while weighting those from the progressively less accurate models increasingly lightly.

The multiple model filter structure is shown in Figure IV-1 (7:132).

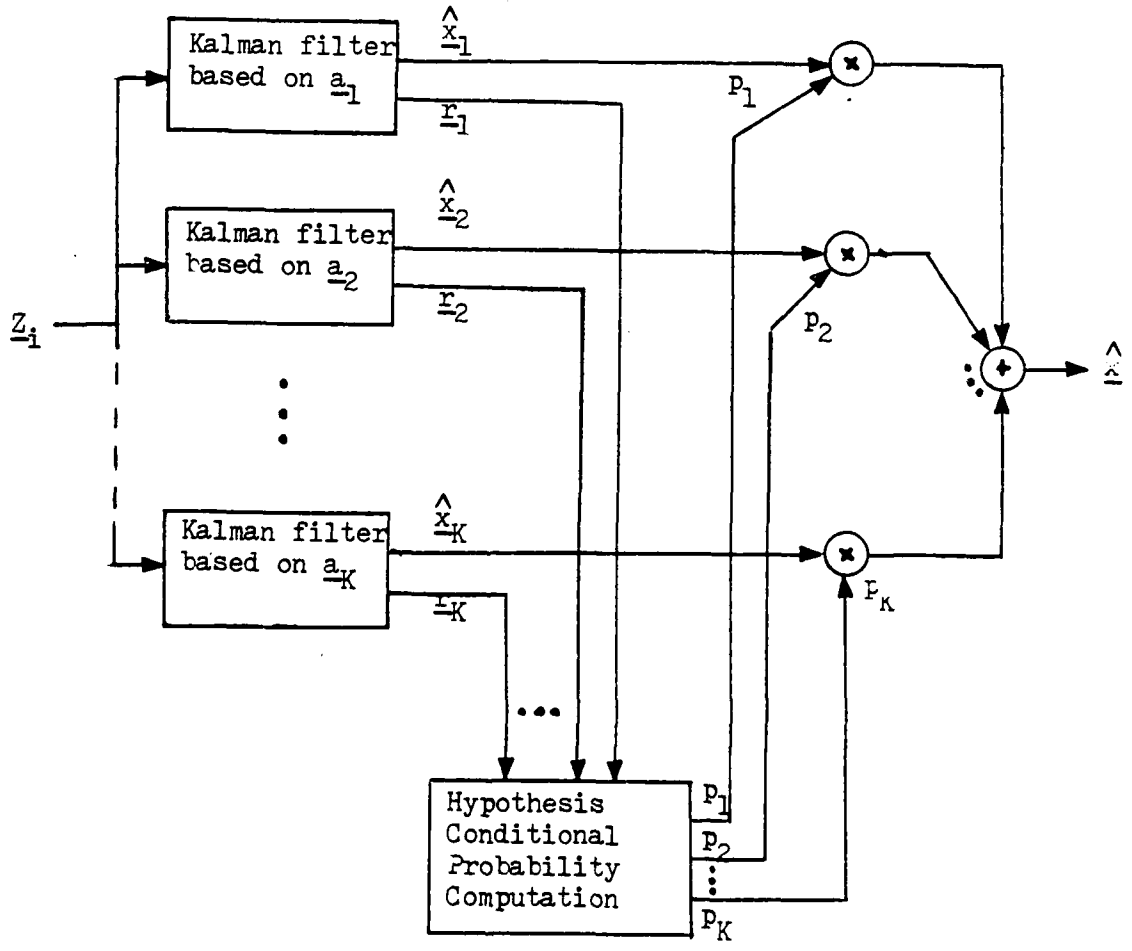


Figure IV-1. Multiple Model Filter Algorithm

As the figure illustrates, the multiple model filter consists of a bank of K independent Kalman filters. The filters are processed in parallel, and they independently produce their own states estimates based on the incoming measurements. When a measurement update is performed at each sample time, the residuals of all filters are used to calculate conditional probabilities which are used to assign the appropriate weighting factors to the estimates of each filter.

These conditional probabilities, called hypothesis conditional probabilities, are the probabilities that the uncertain parameters, $\underline{a}$, have assumed the same values as those modeled by the kth Kalman filter, $\underline{a}_k$, given the measurements received up to that time, for K Kalman filters.

$$p_k(t_i) = \text{prob}\{\underline{a} = \underline{a}_k \mid \underline{Z}(t_i) = \underline{Z}_i\} \qquad (4\text{-}1)$$

where $\underline{a}$ = a random vector which can take on values $\underline{a}_1$ to $\underline{a}_k$

$\underline{a}_k$ = values for the uncertain parameters of the kth Kalman filter where k = 1,2,...,K for K Kalman filters in the filter structure

$\underline{Z}_i$ = measurement time history up to time t_i

$p_k(t_i)$ = weighting factor for the estimates of the kth Kalman filter at time t_i

The conditional probabilities at time t_i can be expressed as (7:131):

$$p_k(t_i) = f_{\underline{z}(ti)|\underline{a},\underline{z}(ti-1)}\ (\underline{z}_i|\underline{a}_k,\underline{Z}_{i-1})\ p_k(t_{i-1}) \Big/ \sum_{j=1}^{K} f_{\underline{z}(ti)|\underline{a},\underline{Z}(ti-1)}\ (\underline{z}_i|\underline{a}_j,\underline{Z}_{i-1})\ p_j(t_{i-1})$$

$$k = 1,2,\ldots,K \quad (4\text{-}2)$$

where the denominator is simply the sum of all computed numerator terms and thus is the scale factor required to insure that the sum of all conditional probabilities is one. This permits the use of the conditional probabilities as the weighting factors of the filters' estimates.

The conditional density functions of (4-2) may be evaluated as:

$$f_{z(ti)|\underline{a},\underline{Z}(ti-1)}\ (\underline{z}_i|\underline{a}_k,\underline{Z}_{i-1}) = (2\pi)^{m/2}\ |\underline{A}_k(t_i)|^{1/2}\ \exp\{\ \cdot\ \} \quad (4\text{-}3)$$

where $\{\ \cdot\ \} = \{-0.5\underline{r}_k^T(t_i)\underline{A}_k^{-1}(t_i)\underline{r}_k(t_i)\}$

$\underline{r}_k$ = residuals of the kth Kalman filter k=1,2,...K

$\underline{A}_k = [\underline{H}_k\underline{P}_k(t_i^-)\underline{H}_k^T + \underline{R}_k]$

As stated earlier, one would expect the filter whose model for $\underline{a}$ most closely resembles the true values for the uncertain parameters to produce the best state estimates. That is, it should have the best behaved residuals, i.e. the most zero mean, white, Gaussian, of covariance equal to the computed value of $\underline{A}_k$ (7:133). It is the size of the residuals relative to the filter's computed estimates of the variance of the residuals errors (via $\underline{A} = [\underline{HPH}^T + \underline{R}]$) that indicates which is the "correct" model for $\underline{a}$.

As illustrated in Figure IV-1, all the filter estimates are combined to form

$$\underline{\hat{x}}(t_i^+) = \sum_{k=1}^{K} \underline{\hat{x}}_k(t_k^+) p_k(t_i) \qquad (4\text{-}4)$$

The conditional covariance of $\underline{\hat{x}}(t_i)$ can be expressed as:

$$\underline{P}(t_i^+) = \sum_{k=1}^{K} p_k(t_i) \{\underline{P}_k(t_i^+) + [\underline{\hat{x}}_k(t_k^+) - \underline{\hat{x}}(t_k^+)] \times [\underline{\hat{x}}_k(t_i^+) - \underline{\hat{x}}_k(t_i^+)]^T\} \qquad (4\text{-}5)$$

It is not absolutely necessary to compute (4-5) in the online algorithm.

As stated earlier, the filter which consistently produces the smallest residuals relative to its own estimates of its errors that is weighted most heavily. Therefore, it is important that there be significant differences between the residuals from this "best" filter and the residuals from the other mismatched filters. Failure to obtain such differences could cause the algorithm to assign inappropriately large probabilites to incorrect models of the uncertain parameter values which will result in poor performance. In terms of implementation, this means that each filter in the algorithm should be tuned for optimal performance when the true values of the uncertain parameters are identical to its model for those parameters. When tuning these filters, one should avoid a "conservative" philosophy, that is, adding large magnitudes of pseudonoise to dynamics. This would

"hide" the inadequacies of the target dynamics model.

Finally, care should be taken when calculating the hypothesis conditional probability values. As can be seen in Equation (4-2), the current value of the conditional probability is the product of the conditional density of Equation (4-3) and the values of the conditional probability at the previous time, divided by the sum of all such products for the K filters. This means that if the conditional probability is allowed to go to zero at any one time, all subsequent values of the conditional probability for that filter will be zero. This will effectively shut off all future contributions from that filter's estimates into the overall multiple model filter estimate. This could reduce the overall filter's ability to respond to future changes in the true parameter values. Consequently, each conditional probability was artificially bounded to keep it from converging to zero (7:135). Another factor which motivated using a lower bound on the conditional probabilities was natural damping effect on the conditional probabilities imposed by the structure of Equation (4-2). Because the current conditional density function is multiplied by the previous value of the conditional probability, there is a certain amount of lag before the filter responds to changes in the true parameter values. By setting the lower bound to a value of 0.01, it was possible to keep the conditional probabilities from converging to zero while simultaneously improving the multiple model filter's ability to respond to

sudden changes in the true parameter values. This value for the bound was chosen based on actual test results of the program. After lower bounding each probability, the resultant probabilities are rescaled so that their sum is kept at one.

4.3 Implementation of the Multiple Model Algorithm

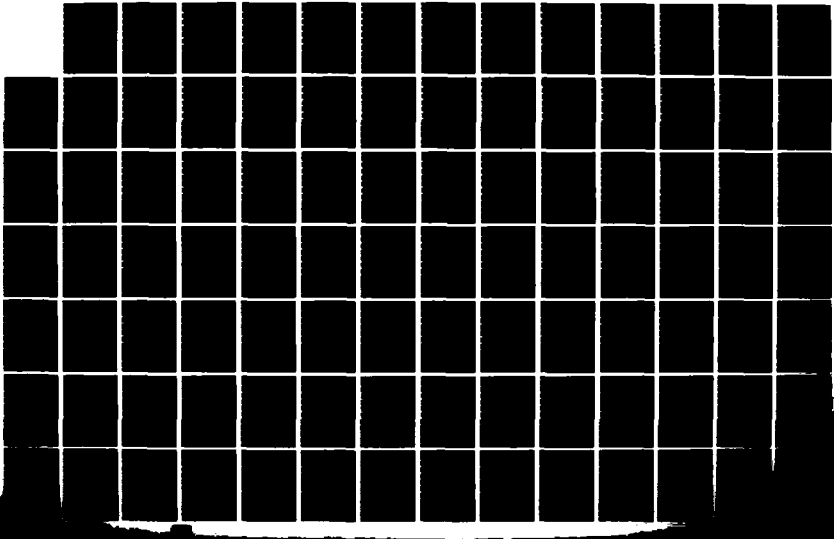
The multiple model algorithm just described was implemented in both the Kozemchak and Millner tracker formulations. While the means of implementation differed because of the inherent differences in the structures of both trackers, certain characteristics are shared by both.

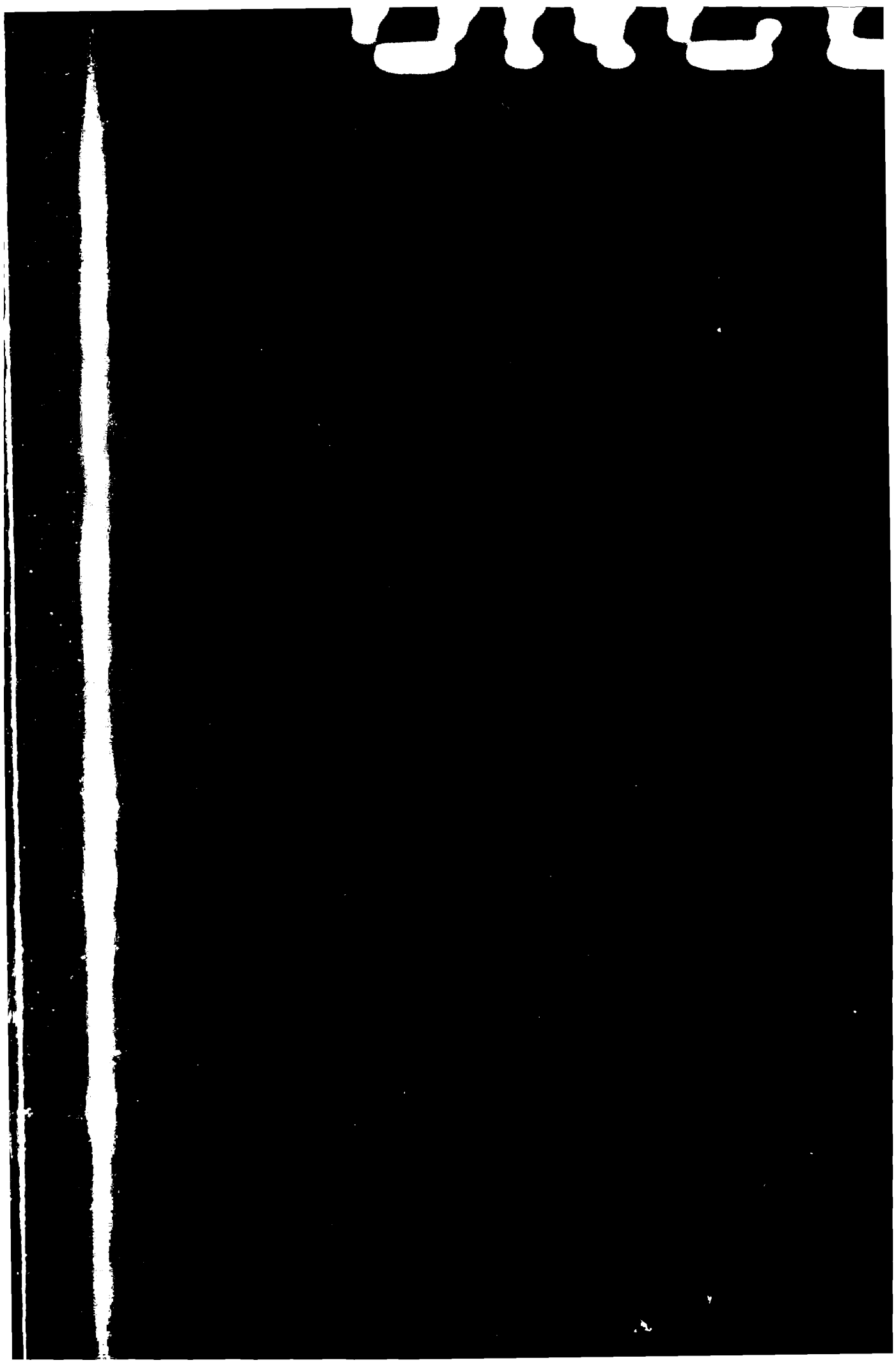
In both filter models, the structure of the original filter and of the truth model as described in Chapters II and III remains unchanged. Also, with the exception of the inclusion of the second filter and the associated algorithms needed to combine the estimates via the multiple model algorithm just described, the structure of each tracker remains as outlined in Chapter I.

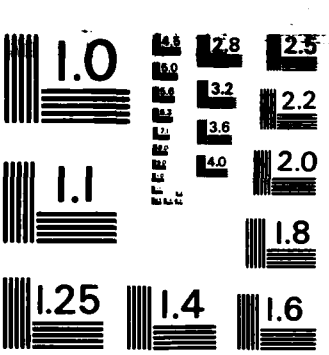
FLIR measurements for the original small field of view are generated as before. Measurements for the larger field of view are generated by taking the 24 x 24 array with the same field of view center as used for the original smaller field of view and averaging the intensities within each 3 x 3 pixel area within that original region, to create an 8 x 8 measurement array for the larger field of view. This averaging of the intensities over each 3 x 3 pixel was done to keep the dimensionality of the filter for the larger

AD-A141 144 ENHANCED TRACKING OF AIRBORNE TARGETS USING MULTIPLE MODEL FILTERING TECH..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. R I SUIZU 2/4

UNCLASSIFIED DEC 83 AFIT/GE/EE/84M-4 F/G 12/1 NL







MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

field of view at the same level as that of the original filter; it's pixels are now 60 μrad-by-60 μrad instead of 20 μrad-by-20 μrad, as originally.

For the larger field of view, the measurement noise of the FLIR was assumed to be spatially uncorrelated because the distance between its pixel centers is now greater than two small (20 μrad) pixels in length. Recall from Chapter II that spatial correlation was modeled as being non-zero only for each pixel and its two closest neighbors in all directions. In addition, the larger field of view makes it reasonable to assume that the target image intensity is essentially zero at the borders. Therefore, the larger field of view was padded with zeros, as opposed to data. Also, all noise contributions to the larger field of view were calculated as the average of the noise terms of the equivalent area for the smaller field of view.

Finally, each filter was tuned for best performance at different target trajectories. The smaller filter was tuned for best performance at very benign trajectories (trajectory 1 as described in Chapter II). The larger filter was tuned to accommodate a highly dynamic target trajectory (trajectory 2 as described in Chapter II with a 20g commanded pull-up maneuver). As explained earlier, this tuning was performed to create significant differences between the residuals of the two filters so that the weighting factors assigned to the estimates of each filter would accurately reflect the suitability of that filter's target dynamics model to the

maneuver being performed at that time.

4.3.1 Implementation in the Kozemchak Tracker Configuration. In this tracker model, the multiple model algorithm was implemented by rescaling the dimensions of the pixels in the second filter. The pixel sizes for the second filter were set to be 60 microradians by 60 microradians, which is a threefold increase in pixel size. Recall from Chapter I that the extended Kalman filter tracker processes the FLIR measurements directly, so the increase in pixel size corresponds to the way the FLIR measurement array for the larger field of view was created in Section 4.3. This rescaling of the pixel size also necessitated that the dynamics model for the second filter also had to be rescaled.

The measurements from the FLIR did not require rescaling except for the aforementioned averaging to account for the larger pixel size. Because of this averaging, the extended Kalman filter measurement vector, $\underline{z}$, remains a 64 x 1 vector, but its components are each the average intensities over a 3 x 3 block of the smaller pixels.

Because $\underline{z}$ is a 64-dimensioned vector, the matrix $\underline{A}$ in Equation (4-3) will be a 64 x 64 matrix. This would require that a full 64 x 64 matrix be performed at every sample time to determine each of the hypothesis conditional probabilities. For a multiple model filter with K Kalman filters, this would correspond to K 64 x 64 matrix inversions at every sample time, assuming each filter is identically dimensioned. As explained in Chapter II, a desire to avoid

doing one such inversion at each sample time required using the inverse covariance method for measurement update of the Kalman filter for the single filter tracker. However, calculation of the $\underline{A}$ matrix is explicitly required in the multiple model filter formulation.

Two approximations were made to alleviate this problem. First, the inverse of the $\underline{A}$ matrix was derived using only the diagonals of $\underline{A}$ (2:24). This approximation was easy to implement and requires only 64 divisions instead of the 32 million operations needed to perform a full inversion. Similarly, the calculation of the determinant of $\underline{A}$, also required in Equation (4-3), requires over 8000 multiplies so it also required an approximation. Since the magnitude of the determinant is independent of the "correctness" of the filter models and it was anticipated that the major differences between the two filters would be in the residuals, the scalar term $(2\pi)^{m/2} |\underline{A}_k|^{1/2}$ terms of Equation (4-3) were ignored (2:24,7:133).

Another approach to providing a value of $\underline{A}^{-1}$ would have been to perform full inversion of those elements of $\underline{A}$ that are associated with the center 4 x 4 region of the field of view. This central region was chosen because it is the area where the centroid of the target is expected to be located (2:24). The $\underline{A}$ matrix would be treated as diagonal or ignored altogether, outside of this region.

Due to the size of the matrices involved, the exponential argument of Equation (4-3) would often exceed the

bounds for the exponential function as implemented on the computer (2:25). A scale factor of .01 was used to bring the argument's magnitude down to acceptable levels. While this scaling reduces the relative ratios between the two filter conditional probabilities, it was deemed acceptable until another means of scaling could be found (2:25). The implemented form of Equation (4-3) was:

$$f_{\underline{z}(ti)|\underline{a},\underline{z}(ti-1)}\ (\underline{z}_i|\underline{a}_k,\underline{z}_{i-1}) = \exp\{-0.005\underline{r}_k^T(t_i) \times \underline{A}_k^{-1}(t_i)\underline{r}_k(t_i)\} \quad (4\text{-}6)$$

For the approximation which uses the 4 x 4 foveal approximation, because of the reduced dimensions of the $\underline{A}$ matrix, no such scaling factor was introduced.

Finally, in Chapter I it was explained that for the extended Kalman filter, the filter state estimates were used to derive the non-linear intensity shape function and the linearized intensity measurement function. This assumed that the appropriate control was applied to FLIR so that the center of the tracking window was located at the predicted target centroid position due to target dynamics as shown in Equation (4-7):

$$\underline{\ell}_{cen}(t_{i+1}^-) = \underline{\ell}_{dyn}(t_{i+1}^-) + \underline{\ell}_{atm}(t_{i+1}^-) \quad (4\text{-}7)$$

where $\underline{\ell}_{cen}(t_{i+1}^-)$ = predicted target centroid location

$\underline{\ell}_{dyn}(t_{i+1}^-)$ = predicted target centroid location due to dynamics. Control is applied to point the sensor toward this spot so it is effectively zeroed out; that is, $\underline{\ell}_{dyn}(t_{i+1}^{-c})$ (dynamic location after control is applied) is zero

$\underline{x}_{atm}(t_{i+1}^-)$ = predicted target centroid location due to atmospheric disturbances

Because the predicted target centroid location due to dynamics is zeroed out, the nonlinear and linearized shape functions are evaluated at the filter's predicted centroid location solely due to atmospherics.

$$(\underline{x}_{dyn}(t_{i+1}^-) + \underline{x}_{atm}(t_{i+1}^-) - \underline{x}_{dyn}(t_{i+1}^-) = \underline{x}_{atm}(t_{i+1}^-) \quad (4-8)$$

However, in the multiple model filter, the FLIR is pointed such that the predicated target centroid position due to target dynamics of the combined estimate is at the center of the field of view. Therefore the intensity shape function and the intensity measurement function for both filters must now be calculated based on the fact that the foveal center is now at some offset distance from the filter's estimate of the location of the target centroid. The nonlinear and linearized intensity functions are now evaluated at:

$$\underline{x}_{dynk}(t_{i+1}^-) + \underline{x}_{atmk}(t_{i+1}^-) - \underline{x}_{dyn(adaptive)}(t_{i+1}^-) \quad (4-9)$$

for the kth filter.

For very severe target maneuvers, the difference between the position estimates of the small field of view filter and those of the multiple model filter become so large that cylindrical shift of the image approaches a complete cycle. At this point, the small field of view filter will diverge. Because of the lower bounding on the

conditional probabilities of each of the filters, this will eventually cause the whole algorithm to diverge. To avoid this condition, the states of the small field of view filter were set to the appropriately rescaled values of the large field of view filter when the magnitude of the shift exceeded 3.0 pixels in magnitude. Its error covariances were set at levels so the filter would undergo another acquisition cycle. The conditional probabilities were kept at their current values to indicate low confidence in the small field of view filter's state estimates.

4.3.2 Implementation in the Millner Tracker Configuration. In the Millner tracker model the rescaling needed to accommodate the larger field of view was accomplished within the correlator, and was transparent to the linear Kalman filter. Recall from Chapter I that the correlator produces "measurements" for the Kalman filter by calculating the offsets from the center of the field of view. Since the correlator provides these estimates in units of pixels, the output of the correlation algorithm for the larger field of view was multiplied by a factor of three prior to being provided to the associated Kalman filter.

Unlike the implementation of the multiple model algorithm in the Kozemchak tracker model, the conditional probabilities can be calculated exactly as outlined in Equations (4-2) and (4-3). This is because $\underline{A}$ is now a 2 x 2 matrix since the measurements to the Kalman filters are offsets from the center of the tracking window.

4.4 Summary

This chapter described the reasons for adopting the multiple model filter algorithm and the structure of the algorithm. It also explained how this algorithm was implemented with the existing tracker configurations.

V. Performance Analysis

5.1 Introduction

This chapter presents the tracking performance of the two tracker formulations against the target trajectories descirbed in Chapter II, after incorporation of the multiple model filter algorithm. The first section of the chapter discusses the figures of merit used to evaluate tracker performance. The next section describes the performance plots generated by each set of computer simulations. The third section of the chapter lists the parameter values assigned to the truth and filter models used in this research. These values were chosen on the basis that they provided the best model of true target behavior and/or they were shown to result in best tracker performance in the studies by Kozemchak (5) and Millner (13). The final section of this chapter discusses the results of the computer simulations using both the figures of merit and the performance plots.

5.2 Derivation of Tracker Statistics

Statistics on tracker performance were gathered using a Monte Carlo analysis technique. Previous studies by Harnly and Jensen (3) and Flynn (2) have shown that a total of 10 Monte Carlo runs will exhibit reasonable convergence of the error statistics to the actual error statistics of an infinite number of runs. Each Monte Carlo run simulated 5 seconds in real time for a total of 150 frames of data at

the 30 Hz sampling rate.

In the tracking problem, one of the quantities of interest is the error committed when estimating the target's true position due to its own dynamics. This error reflects how well the internal filter dynamics model performs against a target performing a wide variety of manuevers. Statistics were also kept on the accuracy of the tracker's estimates of the target centroid. These statistics are of interest because errors in the estimation of the target centroid will affect the accuracy of the estimated target intensity shape function in the extended Kalman filter tracker, and the target reference image in the linear Kalman filter/correlator tracker. Errors in the estimated intensity function are important because this function is used when updating filter estimates each time new information is received from the FLIR. Errors in the target reference image will affect the correlation process in the linear Kalman filter/correlator tracker. This will produce less accurate offset quantities (the pseudo-measurements for the linear Kalman filter), thereby affecting the accuracy of the tracker.

All of the above statistics were kept for instances before and after measurement incorporation. By comparing the errors committed before and after the filter estimates are updated, it is possible to evaluate how well estimates of target position are improved each time information from a frame of data is received.

The sample mean errors of the filter state estimates are calculated as follows:

$$\overline{E_{xd}}(t_i) = 1/N \sum_{k=1}^{N} [x_{dk}(t_i) - \hat{x}_{dfk}(t_i)] = 1/N \sum_{k=1}^{N} e_{xdk}(t_i) \tag{5-1}$$

where $\overline{E_{xd}}(t_i)$ = sample mean error (i.e. ensemble average error over all simulations) in x-dynamics position at time t_i

$\hat{x}_{dfk}(t_i)$ = multiple model filter estimated x-dynamics value at time t_i for simulation k

$x_{dk}(t_i)$ = truth model x-dynamics value at time t_i for simulation k

$e_{xdk}(t_i)$ = error in x-dynamics position at time t_i for simulation k

N = number of Monte Carlo runs

and the sample variance of the error is given by:

$$\sigma_{xd}^2(t_i) = 1/(N-1) \sum_{k=1}^{N} e_{xdk}^2(t_i) - [N/(N-1)] \overline{E_{xd}}^2(t_i) \tag{5-2}$$

where the quantities are as defined above. The two equations, (5-1) and (5-2), may be generalized to perform sample mean error and variance calculations for the errors committed when estimating the y-dynamics position, and the x- and y- centroid location coordinates. These errors are expressed in FLIR image plane coordinates and describe offset from the center of the sensor field of view. The units of the errors are pixels, with each pixel being 20 rads in length.

In addition to averaging the errors over all Monte Carlo runs, time averaging was performed on the mean errors and standard deviations. This temporal averaging was conducted from t=0.5 seconds to t=2.0 seconds for trajectory 1 evaluations, and from t=3.5 seconds to t=5.0 seconds for all other trajectories. The earlier time frame for trajectory 1 evaluations was selected because it allowed the filter transients to die out while avoiding the minimum range/maximum passing rate condition that occurs near the end of the simulation. In this condition, any filter tuned for a benign target trajectory begins to exhibit markedly degraded performance. The later time frame chosen for the other trajectories allows the filter transients due to manuevers initiated at t=2.0 seconds to die out completely before time averaging begins.

This time averaging allows presentation of the data in a compact, tabular form. However, temporal averaging can also result in misleading figures of merit. For instance, if the errors should follow a ramp function from negative to positive values over the period of temporal averaging, a misleading sample mean error of approximately zero will be the result. This figure could lead to an erroneous assumption that an unbiased estimate was being generated by the filter. Therefore, care should be taken before making sweeping generalizations based only on the figures presented in the tables. For this reason, the performance plots of the simulations are included in Appendices A, B, and C.

These plots are grouped according to trajectory type and according to severity of the maneuver within those groups. The performance plots for trajectory 1 evaluations are followed by those for trajectory 2, trajectory 3, and son on. Within those groups, the plots for a 2g pull-up maneuver are followed by those for the 10g pull-up maneuver, and then the 20g maneuver. Still further, the plots of the multiple model filter are followed by those of the small field of view filter, and then those of the large field of view filter. Appendix A contains the plots for the linear Kalman filter/correlator tracker. Appendix B contains the plots for the extended Kalman filter tracker using the Gauss-Markov target acceleration model. Appendix C contains the plots for the same tracker as Appendix B, but these results are for the constant turn rate target acceleration model.

5.3 Performance Plots

As mentioned in the previous section, performance plots of the simulations are generated to prevent possible misinterpretations of the results caused by the temporal averaging of the statistics. These performance plots are of the x- and y- dynamics mean errors, and the x- and y- centroid mean errors; plus and minus the standard deviation of the respective errors. Because of the large number of cases run during this research, only the x- and y- dynamics mean error (plus and minus one standard deviation) plots at both before and after measurement incorporation are included in this

document. This is due to the fact that: (1) estima true target states x_d and y_d are of primary importan tracking; and (2) it is in fact easier to estimat position of the apparent target centroid than to ide the individual components of Equation (2-1). The num plots was further pared down by including only those which illustrated important trends. For example, most involving target maneuver will have plots only for t axis, which was the direction of the maneuver.

Examples of the types of performance plots gen are Figures V-1 and V-2. These are plots of the y-dy mean error plus and minus one standard deviation. these figures, it is evident that the maneuver was ini at t=2.0 seconds. At that point, a dramatic incre mean tracker error is exhibited. It can also be see it takes the filter some finite amount of time to re Figure V-1 is a plot of the y-dynamics error statist the time prior to measurement incorporation, or t minus" error. Figure V-2 is a plot of the same qua but at the time after measurement incorporation, or t plus" error. To illustrate how well the filter improv target position estimates each time measurement infor is received from the FLIR, note the mean peak err Figures V-1 and V-2. The mean peak error is approxi -7.0 pixels for the minus time, and is only -3.8 pixe the plus time. Pixels are defined as before, wit pixel measuring 20 μrad in length. The values of th

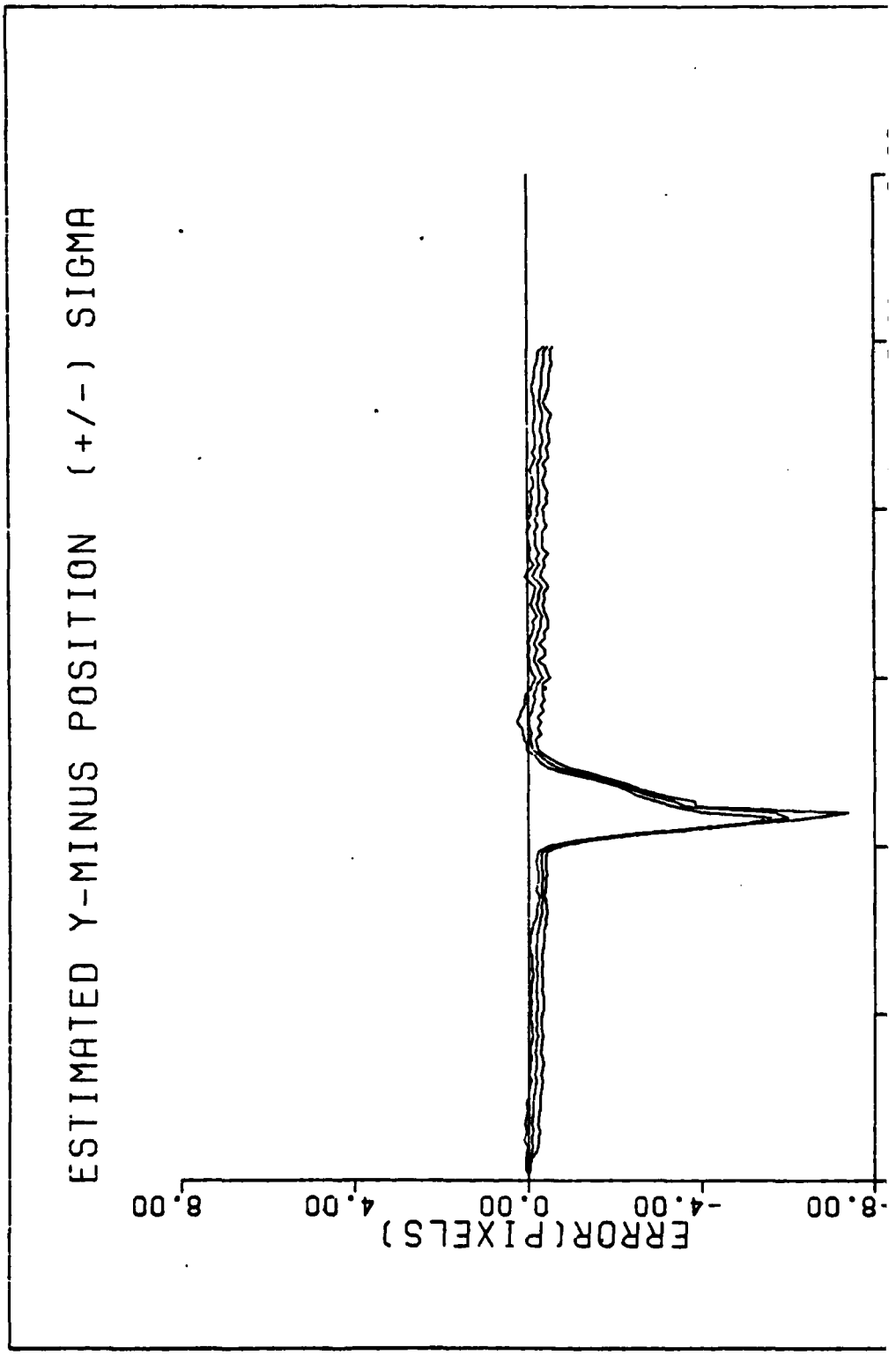


Figure V-1. Y-Minus Error, 20g Pull-up Maneu

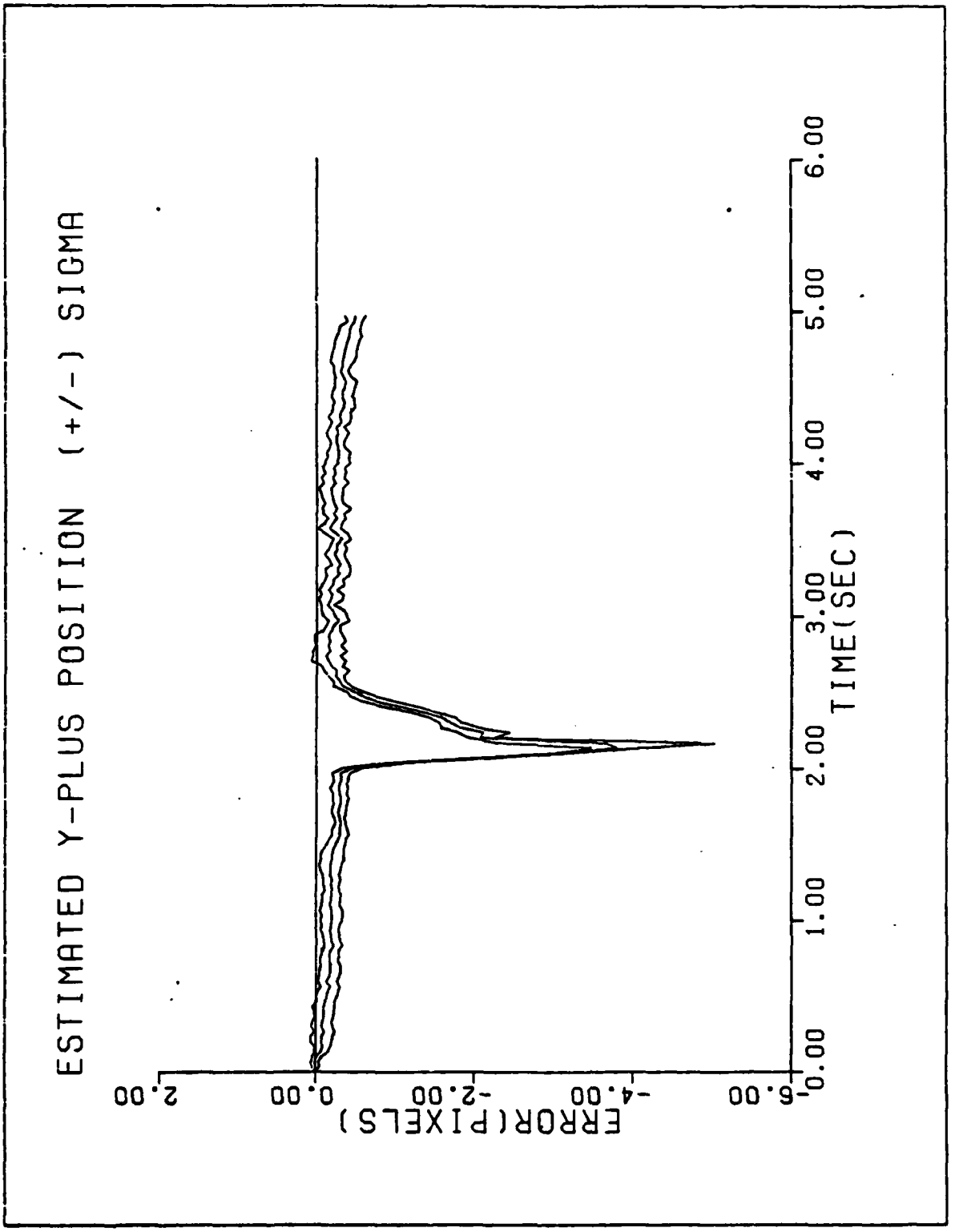


Figure V-2. Y-Plus Error, 20g Pull-up Maneuver

error and standard deviation are generated by applying Equations (5-1) and (5-2) to the data at each sample time for each Monte Carlo run.

5.4 Parameter Values

In order to allow direct comparison of the results obtained for the extended Kalman filter tracker and the linear Kalman filter/correlator tracker, both trackers were presented with as identical as possible truth model representations of the target. Therefore, the truth models of both the extended Kalman filter and the linear Kalman filter/correlator trackers were set so they possessed identical descriptions of target shape and intensity, as well as identical models for background and FLIR measurement noises, atmospheric jitter, etc. On the other hand, the parameter values used in the filter models for the respective trackers will vary somewhat with the tuning done for each tracker.

5.4.1 Truth Model Parameters. For all simulations, the initial inertial parameters of the target in the inertial reference frame were:

$$\begin{aligned} \text{Inertial position:} \quad & x = 5000 \text{ m} \\ & y = 500 \text{ m} \\ & z = 20000 \text{ m} \end{aligned}$$

$$\begin{aligned} \text{Inertial velocity:} \quad & v_x = -1000 \text{ m/s} \\ & v_y = 0 \\ & v_z = 0 \end{aligned}$$

$$\begin{aligned} \text{Inertial acceleration:} \quad & a_x = 0 \\ & a_y = 0 \\ & a_z = 0 \end{aligned}$$

All trajectories described in Chapter II start from these initial conditions. The input measurement variance, which includes both the background and FLIR noises, was set to a value of one. This parameter value was expressed in terms of $(\text{intensity})^2$ units, which are arbitrarily chosen units used to indicate the strength of the image received by the FLIR. The maximum intensity of each target hot-spot was given a value of 20. The resulting signal-to-noise (SNR) of all simulations is 20, which is defined by the following relationship for this application:

$$\text{SNR} = \frac{\text{(maximum signal intensity)}}{\text{(rms background and FLIR noise intensity)}} \qquad (5\text{-}3)$$

This value for the SNR is representative of realistic tracking scenarios (3). The variance of the atmospheric jitter was set to 0.2 pixels^2. This value may be somewhat low compared to the true level of atmospheric jitter in the real world so it may be advisable to investigate the sensitivity of the algorithm to different levels of atmospheric jitter in future studies. Finally, the multiple hot -spot target were defined using circular intensity contours with a glint dispersion parameter of 2.0 pixels^2.

5.4.2 Data Processing Parameters. In the data processing algorithms of both tracker types, a number of parameters could be varied to alter tracker performance. One such parameter determines the nature of the padding done to the 8 x 8 tracking window discussed in Chapter I. Recall

that a 24 x 24 array is processed instead of the 8 x 8 field of view to generate the FFT's. A common engineering practice would be to pad the field of view by surrounding it with a border of 8 rows and columns of zeros. However, as stated earlier, this would introduce artificial edge effects if the target intensity function is not very close to zero at the borders of the 8 x 8 tracking window. Since measurement data of the regions outside the tracking window was available, there was the luxury of being able to pad the tracking window of the smaller field of view with real, but noise-corrupted data, instead of zeros. On the other hand, the increased size of the larger field of view motivated padded it with zeros at all times since it is better to pad with zeros versus the noise value, if the signal has gone to zero at the borders of the tracking window.

High frequency spatial frequency filtering could also be performed for both tracker types when deriving the target intensity profile. This can be accomplished by zeroing out a specified number of high frequency components within the Fourier transform of the image. For a target whose intensity is slowly changing in the spatial domain, this type of filtering will generally enhance tracker performance. Conversely, for a rapidly changing target intensity function, this type of filtering will produce errors in the target reference image and degrade tracker performance (13:106). In both tracker configurations, it was inappropriate to perform high frequency filtering for the larger field of

view since, with pixel scaling, the large field of view signal spatial frequency content goes to three times the highest frequency as seen in the original field of view.

Finally, the relative weighting parameter used in performing exponential smoothing can be varied to account for a rapidly changing target intensity profile. This parameter was set to 0.1 for both tracker formulations since it was found to yield best performance in previous studies (5,13). This indicates that the target intensity function is expected to be slowly varying relative to the 30 Hz sampling rate.

5.4.3 Filter Parameter Values. As stated in Chapter IV, each filter in both tracker types was tuned for optimum performance at a specified degree of target motion. The smaller field of view filters were tuned to achieve best performance for constant velocity trajectories. The large field of view filters were tuned for best performance for the 20g pull-up maneuver. During this research, it was found that explicitly tuning the larger field of view for a 20g pull-up manuever made it incapable of maintaining lock on a constant velocity target. Therefore, this filter was tuned for best performance for a 20g pull-up maneuver while still being able to track a constant velocity target. Because this study was intended to be a feasibility study for the multiple model adaptive filter algorithm, no investigations were made to determine the increase in rms error caused by the need to maintain lock on a constant velocity

target. In a tracking scenario, this philosophy makes sense since the wider aperture would also be used for target acquisition, where the targets would be at long-range and exhibiting very benign behavior. The performance achieved from the individual filters under these tuning conditions will be presented in Table V-1 and associated figures.

For all filters, atmospheric jitter was modelled with an assumed correlation time of 0.07 seconds. This value was chosen based on the results of previous research efforts (3,12,14).

For the extended Kalman filter tracker, the values for the parameters for the smaller field of view filter were selected on the basis of those values which gave best performance for the benign target trajectories in Kozemchak's (5) study. A filter time correlation constant for filter acceleration dynamics, using the first-order Gauss-Markov model, was set to 1.5 seconds, with the assumed target acceleration white noise strength set to 300 $pixels^2/sec^5$. As before, pixels are defined to be 20 μrads in length. Except where explicitly stated, this definition of a pixel will hold at all times. Parameters for the larger field of view will also be expressed in these pixel units even though the actual picture elements of the larger field of view measure 60 μrads in length. On the basis of tuning runs performed, the time correlation constant for filter acceleration dynamics was set to 1.3 seconds for the Gauss-Markov model. The target acceleration white noise strength was set

to 5000 pixels^2/sec^5 for both the constant turn rate and Gauss-Markov target acceleration models for the larger field of view filter. The similarity of the values for the two disimilar acceleration models is due in part to the need to maintain lock on a constant velocity target.

For the linear Kalman filter/correlator tracker, Millner's (13) results indicated that a dynamic correlation time of 3.5 seconds and an assumed target acceleration white noise strength of 150 pixels^2/sec^5 achieved best performance for the small field of view filter. The large difference in the dynamics correlation time for the two tracker types indicates that perhaps the larger correlation time for the linear Kalman filter/correlator tracker could be reduced to be more consistent with the value of the same parameter for the extended Kalman filter tracker. This was not investigated during this research and should be explored in any future study. Test runs of the larger field of view filter indicated it performed best for the 20g pull-up maneuver when these parameters were set to 1.5 seconds and 2000 pixels^2/sec^5 respectively.

Before presenting the performance capabilities of the individual filters at their tuned-for conditions, a narrative on the organization and presentation of the information would be helpful. First, each case, or simulation is uniquely identified using a mnemonic code. This code is described in Figure V-3. The figures of merit presented in the tables are defined as:

$\bar{x}_e$ = time-average of the mean error for the true position in the x-direction from time t=3.5 seconds to t=5.0 seconds (t=0.5 seconds to t=2.0 seconds for trajectory 1 targets) at time minus and plus (similarly for $\bar{y}_e$)

$\overline{\sigma x}_e$ = time-average of the standard deviation of the error for the true position in the x-direction from the time t=3.5 seconds to t=5.0 seconds (t=0.5 seconds to t=2.0 seconds for trajectory 1 targets) at times minus and plus (similarly for $\overline{\sigma y}_e$)

$\overline{cx}_e$ and $\overline{\sigma cx}_e$ = errors as degined above for the centroid position (similarly for cy_e and cy_e)

The performance capabilities of the respective filters are presented in Table V-1. The first three cases presented in the table are those of the small field of view filters. Recall that this filter was tuned for the constant velocity trajectory. The performance plots of the x-dynamics error for these cases are Figures A-2, B-2, and C-2, respectively.

The Millner linear filter/enhanced correlator tracker exhibits lower sample mean errors but with larger standard deviations. From the performance plots for the extended Kalman filter tracker cases, the poor tuning of the small field of view filter is clearly evident. The estimates show a drifting behavior that indicates that the filter dynamic driving noise was set too low. Unfortunately, this poor tuning condition (based on tuning values established in reference [5]) was not noticed until late in this research. At that point, while it would have still been possible to perform a total retuning of the filter, for this feasibility study, there was some concern that increasing the level of the filter dynamics driving noise would reduce the differ-

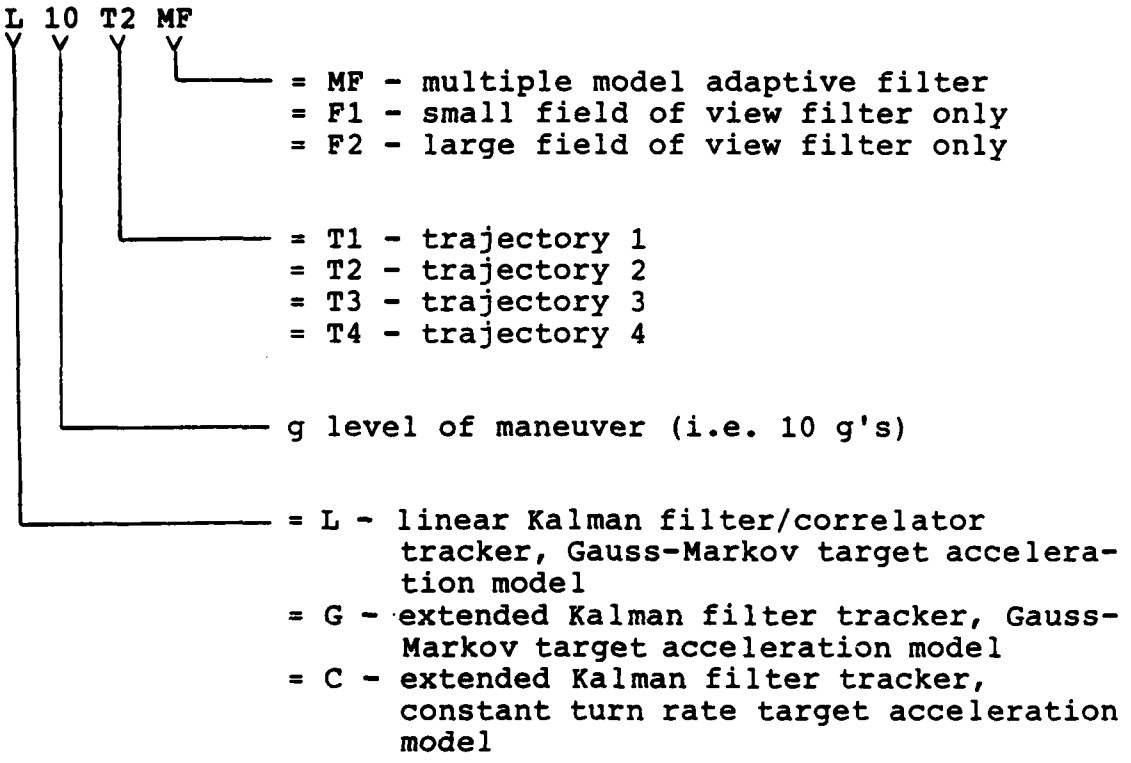
TABLE V-1

INDIVIDUAL FILTER EVALUATIONS

CASE	$\overline{x}_e(-)$ / $\overline{\sigma x}_e(-)$	$\overline{x}_e(+)$ / $\overline{\sigma x}_e(+)$	$\overline{cx}_e(-)$ / $\overline{\sigma cx}_e(-)$	$\overline{cx}_e(+)$ / $\overline{\sigma cx}_e(+)$	$\overline{y}_e(-)$ / $\overline{\sigma y}_e(-)$	$\overline{y}_e(+)$ / $\overline{\sigma y}_e(+)$	$\overline{cy}_e(-)$ / $\overline{\sigma cy}_e(-)$	$\overline{cy}_e(+)$ / $\overline{\sigma cy}_e(+)$
LT1F1	.046/ .180	.064/ .160	.080/ .177	.108/ .062	-.015/ .170	-.020/ .152	-.012/ .181	-.020/ .057
GT1F1	-.236/ .135	-.195/ .124	-.180/ .133	-.107/ .079	-.266/ .118	-.270/ .106	-.263/ .126	-.271/ .053
CT1F1	.208/ .141	.184/ .129	.185 .138	.145/ .084	-.209/ .123	-.210/ .112	-.203/ .131	-.206/ .067
L20T2F2	.583/ .237	.355/ .193	.386/ .239	-.053/ .113	-.264/ .250	.036/ .214	-.040/ .232	.398/ .129
G20T2F2	.354/ .198	.186/ .161	.241/ .171	.015/ .085	-.884/ .204	-.674/ .181	-.766/ .186	-.484/ .127
C20T2F2	.084/ .203	.073/ .162/	.060/ .174	.045/ .087	-.117/ .194	-.123/ .162	-.127/ .175	-.135/ .097

Each simulation is uniquely identified with a mnemonic code. This code will consist of up to eight characters and describes the tracker formulation used, target acceleration model, trajectory type, and whether the multiple model, the small field of view filter, or the large field of view filter was used in the simulation.

The code generally follows this pattern:



Special initial codes:

LA, GA, CA	tracker with ad hoc changes to conditional probabilites
LS	linear Kalman filter/correlator tracker, SNR = 10

Figure V-3. Simulation Mnemonic Codes

ences between the large and small field of view filter models, thus making it difficult for the multiple model adaptive filter algorithm to select the correct filter model to match current target behavior. This lack of significant differences hampered the early study on the multiple model filter algorithm by Flynn(2). Due to this poor tuning of the smaller field of view filter for the extended Kalman filter tracker, the Millner tracker formulation also exhibits smaller sample rms errors for the benign trajectory. However, in a laser weapon system, it may be more important to minimize the area painted by the laser so it may be more desirable to have smaller standard deviations than small offset errors; i.e. small standard deviations may be more essential than small mean errors. One final characteristic evident in these performance plots is the increase in the mean errors during the last 0.5 seconds of each simulation. This increase is due to the minimum range/maximum passing rate condition alluded to earlier in this chapter.

The second group of cases cited in Table V-1 are those of the large field of view filters for the 20g pull-up maneuver. The performance plots for these cases are Figure A-10, B-10, and C-10. Only the y-dynamics error plots are presented since the maneuver took place in that direction. As before, the extended Kalman filter tracker has much smaller sample standard deviations. However, that tracker formulation, using the contant turn rate acceleration model, demonstrates far superior performance. As shown in the

performance plots, its peak mean errors are smaller and the bias of the estimate once the filter transients have settled down is very small. The performance plots of the Millner tracker formulation (Figure A-10) show similar mean peak errors with and unbiased estimate after the transients have died out. But comparing the length of time it takes the filter to recover from the manuever shows that the extended Kalman filter tracker using the constant turn rate acceleration model has a much shorter transient time. Figure B-10, which is of the same tracker formulation but with the Gauss-Markov model shows the same good transient behavior but with larger peak mean errors. The figure also graphically illustrates the large bias that appears in Table V-1. This large rms error indicates the inadequacies of using the first-order Gauss-Markov target acceleration model for tracking highly dynamic targets.

5.5 Tracker Performance Against Target Trajectories

Both multiple model tracker formulations were evaluated against the target trajectories described in Chapter II. For convenience, they are summarized here. Trajectory 1 is a constant velocity trajectory which maintains the initial velocity throughout the simulation. Trajectory 2 is a constant g pull-up maneuver. It begins the same as trajectory 1, but the target initiates a constant-speed, constant-g pull-up maneuver two seconds into the simulation and maintains this maneuver until the end of the simulation. The cases studied included 2, 10, and 20g pull-up maneuvers.

Trajectory 3 is similar to trajectory 2, but inste
continuing the pull-up maneuver until the end of the s
tion, it terminates the pull-up maneuver 3.5 second
the simulation. At this point, the inertial velocity
existing at that time are propagated until the end
simulation at 5 seconds. Trajectory 4 is begun a
previous maneuvers, but instead of performing a pu
maneuver, it executes a constant-g turn toward the t
until the end of the simulation.

5.5.1 Evaluations Using Trajectory 1. For this t
tory, Monte Carlo runs were performed to evaluate t
performance using both filters in the multiple model
tive algorithm; only the small field of view filte
only the large field of view filter. Recall from Chap
that a claim was made that simply enlarging the t
field of view for all applications would result in
tracker performance at benign trajectories relative t
achieved using the small field of view filter.

Table V-2 presents the results of these simul
using the multiple model adaptive algorithm and on
large field of view filter for both tracker types. Re
that the results of the simulations using only the
field of view filter were presented in Table V-1.
corresponding performance plots are Figures A-1 and A-
and B-3; and C-1 and C-3.

The first two cases in the table are for the Mi
tracker formulation. The first of these is for the mu

model adpative algorithm, the second is for the larg of view filter only. As expected, the larger field filter, which was tuned to be able to handle a 20g maneuver, performed poorly against a benign target tory relative to the original filter presented in Ta which was expressly tuned for good performance agai type of target behavior. Comparing the figures of m the multiple model adaptive filter and those of t field of view filter reveals values that are very indicating that the multiple model algorithm is c weighting the estimates of the small field of vie heavily, thus maintaining the desired high resol benign target trajectories. This heavy weightin small field of view filter also manifests itsel similarity of the performance plots for the multip and small field of view filters (Figures A-1 and A-: tively).

Results for the extended Kalman filter track rithm are similar though less conclusive due to tuning of the small field of view filter. In this c larger field of view filter exhibits much lower sam and rms errors than the small field of view filte ever, its standard deviations are much higher. The acteristics are evident in the performance plots, F. 3 and C-3. It should also be noted that the multip filter figures of merit resemble those of the small view filter (as with the linear Kalman filter/co

TABLE V-2

TRAJECTORY 1 EVALUATIONS

CASE	$\overline{x}_e(-)$ / $\overline{\sigma x}_e(-)$	$\overline{x}_e(+)$ / $\overline{\sigma x}_e(+)$	$\overline{cx}_e(-)$ / $\overline{\sigma cx}_e(-)$	$\overline{cx}_e(+)$ / $\overline{\sigma cx}_e(+)$	$\overline{y}_e(-)$ / $\overline{\sigma y}_e(-)$	$\overline{y}_e(+)$ / $\overline{\sigma y}_e(+)$	$\overline{cy}_e(-)$ / $\overline{\sigma cy}_e(-)$	$\overline{cy}_e(+)$ / $\overline{\sigma cy}_e(+)$
LT1MF	.052/ .194	.067/ .172	.082/ .179	.110 .063	-.001/ .063	-.005/ .157	.003/ .188	-.004/ .078
LT1F2	.098/ .297	.120/ .245	.125/ .295	.157/ .218	-.017/ .289	-.020/ .246	-.011/ .299	-.014/ .228
GT1MF	-.223/ .134	-.185/ .121	-.170/ .129	-.101/ .069	-.241/ .114	-.246/ .102	-.239/ .122	-.247/ .046
GT1F2	-.006/ .191	.008/ .155	.008/ .168	.028/ .092	.015/ .162	.013/ .128	.020/ .158	.018 .075
CT1MF	.167/ .141	.145/ .128	.147/ .135	.110/ .076	-.208/ .121	-.210/ .109	-.202/ .127	-.205/ .058
CT1F2	.037/ .194	.026/ .156	.036/ .170	.022/ .093	.013/ .166	.012/ .129	.019/ .161	.018/ .075

tracker) despite the fact the larger field of view filter has smaller rms error values. This characteristic is confirmed by comparing Figures B-1 and B-2, and Figures C-1 and C-3. This indicates that the multiple model algorithm was still able to choose the correct filter model even in this pporly tuned case. those of the larger field of view filter even though it has smaller sample rms error values. This indicates that the multiple model algorithm was able to choose the correct filter model even in this poorly tuned case.

5.5.2 Evaluations Using Trajectory 2. Tables V-3, V-4, and V-5 present the results of the simulations for 2, 10, and 20g constant speed, constant-g pull-up maneuvers respectively. The corresponding performance plots are Figures A-4 thru A-10, B-4 thru B-10, and C-4 thru C-10.

Table V-3 presents the results for a 2g pull-up maneuver. Along with the results for the multiple model adaptive filter algorithm, are those for the individual filters. Generally speaking, the linear Kalman filter/correlator tracker produces lower sample mean errors with much larger standard deviations than the extended Kalman filter tracker configuration. In this instance, the sample rms errors of the multiple model filter in the Millner tracker formulation are also much smaller. This is again largely due to the poor tuning of the smaller field of view filter for the Kozemchak tracker formulation. As seen in the performance plots and the figures of merit, the performance of the

TABLE V-3

TRAJECTORY 2 EVALUATION (2G)

CASE	$\overline{x}_e(-)$ / $\overline{\sigma x}_e(-)$	$\overline{x}_e(+)$ / $\overline{\sigma x}_e(+)$	$\overline{cx}_e(-)$ / $\overline{\sigma cx}_e(-)$	$\overline{cx}_e(+)$ / $\overline{\sigma cx}_e(+)$	$\overline{y}_e(-)$ / $\overline{\sigma y}_e(-)$	$\overline{y}_e(+)$ / $\overline{\sigma y}_e(+)$	$\overline{cy}_e(-)$ / $\overline{\sigma cy}_e(-)$	$\overline{cy}_e(+)$ / $\overline{\sigma cy}_e(+)$
L2T2MF	.226/ .182	.197/ .164	.150/ .189	.089/ .069	-.157/ .225	-.118/ .209	-.099/ .228	-.020/ .151
L2T2F1	.230/ .175	.197/ .159	.146/ .185	.078/ .059	-.187/ .166	-.146/ .152	-.123/ .180	-.038/ .069
L2T2F2	.209/ .303	.203/ .251	.180/ .307	.171/ .218	.118/ .317	.152/ .274	.135/ .326	.185/ .248
G2T2MF	.175/ .122	.165/ .109	.146/ .151	.129 .099	-.802/ .129	-.737/ .119	-.725/ .135	-.609/ .083
G2T2F1	.198/ .124	.189/ .112	.169/ .154	.152/ .103	-.796/ .129	-.731/ .119	-.718/ .136	-.601/ .086
G2T2F2	.051/ .196	.046/ .159	.030/ .171	.024/ .083	-.006/ .174	.017/ .139	.001/ .164	.033/ .076
C2T2MF	.409/ .128	.384/ .114	.359/ .157	.315/ .105	-.435/ .145	-.432/ .134	-.439/ .149	-.433/ .105
C2T2F1	.436/ .130	.410/ .116	.386/ .158	.341/ .108	-.422/ .149	-.419/ .138	-.427/ .153	-.421/ .111
C2T2F2	.070/ .201	.060/ .160	.047/ .173	.034/ .085	.039/ .178	.039/ .141	.033/ .167	.032/ .077

multiple model adpative filter more closely resembles that of the small field of view filter since its constant velocity model more closely resembles the 2g pull-up maneuver than does the 20g pull-up maneuver model. The performance plots also show that the bias for the extended Kalman filter tracker using the Gauss-Markov model is very large compared to the biases for the other cases. Finally, the performance plots for the large field of view filter (Figures A-6, B-6, and C-6 respectively) show that, because of the large filter target acceleration white noise strength, the maneuver is not as evident as in the plots of the small field of view filter.

Tables V-4 and V-5 respectively, present the results for 10 and 20g pull-up maneuvers. The performance plots for these cases are Figures A-7 thru A-10, B-7 thru B-10, and C-7 thru C-10. Only plots for the y-axis errors are included since that is the direction of target motion. Furthermore, only cases using the multiple model filter algorithm and only the large field of view filter are presented here as the small field of view filter was completely unable to maintain lock on these targets. For these highly dynamic cases, the extended Kalman filter tracker exhibits much smaller sample rms errors than the linear Kalman filter/correlator tracker. For both tracker types the multiple model filter has slightly worse figures of merit than those of the large field of view filter by itself. This is due to the lower bounding of the conditional probabilities discussed in

TABLE V-4

TRAJECTORY 2 EVALUATIONS (10G)

CASE	$\overline{x}_e(-)$ / $\overline{\sigma x}_e(-)$	$\overline{x}_e(+)$ / $\overline{\sigma x}_e(+)$	$\overline{cx}_e(-)$ / $\overline{\sigma cx}_e(-)$	$\overline{cx}_e(+)$ / $\overline{\sigma cx}_e(+)$	$\overline{y}_e(-)$ / $\overline{\sigma y}_e(-)$	$\overline{y}_e(+)$ / $\overline{\sigma y}_e(+)$	$\overline{cy}_e(-)$ / $\overline{\sigma cy}_e(-)$	$\overline{cy}_e(+)$ / $\overline{\sigma cy}_e(+)$
L10T2MF	.338/ .458	.269/ .432	.250/ .466	.144/ .419	-.089/ .747	.084/ .740	.046/ .729	.307/ .706
L10T2F1	.148/ .282	.085/ .232	.077/ .271	-.014/ .177	.337/ .308	.508/ .280	.461/ .298	.711/ .234
G10T2MF	.131/ .197	.084/ .160	.086/ .171	.023 .086	-.232/ .171	-.107/ .138	-.164/ .160	.006/ .073
G10T2F2	.129/ .196	.083/ .159	.085/ .171	.023/ .084	-.175/ .172	-.052/ .138	.111/ .161	.055/ .074
C10T2MF	.109/ .179	.097/ .147	.078/ .155	.058/ .063	-.203/ .167	-.221/ .139	-.211/ .158	-.232/ .081
C10T2F2	.074/ .201	.064/ .160	.050/ .173	.037/ .085	.055/ .178	.052/ .141	.047/ .167	.044/ .078

TABLE V-5

TRAJECTORY 2 EVALUATIONS (20G)

CASE	$\overline{x}_e(-)$ / $\overline{\sigma x}_e(-)$	$\overline{x}_e(+)$ / $\overline{\sigma x}_e(+)$	$\overline{cx}_e(-)$ / $\overline{\sigma cx}_e(-)$	$\overline{cx}_e(+)$ / $\overline{\sigma cx}_e(+)$	$\overline{y}_e(-)$ / $\overline{\sigma y}_e(-)$	$\overline{y}_e(+)$ / $\overline{\sigma y}_e(+)$	$\overline{cy}_e(-)$ / $\overline{\sigma cy}_e(-)$	$\overline{cy}_e(+)$ / $\overline{\sigma cy}_e(+)$
L20T2MF	.537/ .441	.310/ .421	.330/ .454	-.007/ .412	-.243/ .450	-.026/ .436	-.032/ .436	.367 .400
L20T2F1	.583/ .237	.355/ .193	.386/ .239	-.053/ .113	-.264/ .250	.036/ .214	-.040/ .232	.398/ .129
G20T2MF	.356/ .200	.186/ .164	.243/ .172	.015/ .087	-.923/ .181	-.710 .155	-.805/ .166	-.519/ .097
G20T2F2	.354/ .198	.186/ .161	.241/ .171	.015/ .086	-.884/ .204	-.674 .181	-.766/ .186	-.484/ .127
C20T2MF	.116/ .187	.104/ .152	.086/ .159	.067/ .069	-.285/ .160	-.300/ .129	-.297/ .150	-.316/ .064
C20T2F2	.084/ .203	.073/ .162	.060/ .174	.045/ .087	-.117/ .194	-.123/ .162	-.127/ .175	-.135/ .097

Chapter IV. This results in having at least part of the state estimates being based on the estimates of a filter with a totally inappropriate model for the current target behavior. The lower bounding could cause a problem if at any time either of the filters in the bank should diverge. During the course of this research, it was found that a higher setting of the lower bound allowed the multiple model filter to apply heavy weight to the correct filter much more quickly, at the expense of a heavier weight on a totally inappropriate model after the transients have died out.

From the performance plots of these cases, it can be seen that the extended Kalman filter tracker exhibits better transient behavior than the other tracker formulation. This includes a shorter transient in response to a step change in the target truth model acceleration and lower mean peak errors as well. This is due to the non-linear nature of the algorithm as well as the fact that the extended Kalman filter tracker operates directly with the raw FLIR data while the other tracker type receives only position offset information from the enhanced correlator.

In the Millner tracker formulation, the multiple model as well as the single filter cases performed more poorly for a 10g pull-up maneuver than the more severe 20g pull-up maneuver. This is because the large field of view filter was tuned for best performance for a 20g pull-up maneuver while no equivalent filter was tuned for a 10g pull-up maneuver. In other words, the 10g maneuver was a case for

which no filter model existed. However, no such behavior was observed for the extended Kalman filter tracker using either target acceleration model. This could be due to the poor tuning of the small field of view filter or perhaps the nonlinearity of the filter itself. These issues should be investigated in future studies.

At the 20g pull-up maneuver, use of the constant turn rate target acceleration model results in superior performance over all other cases. This follows since this acceleration model has been shown to be a better model of realistic, highly-dynamic, target maneuvers than the Gauss-Markov acceleration model.

5.5.3 Evaluations Using Trajectory 3. Tables V-6, V-7, and V-8 present the results of the simulations for trajectory 3 targets performing 2, 10, and 20g pull-up and continuation maneuvers. The plots for these cases are Figures A-11 thru A-17, B-11 thru B-17, and C-11 thru C-17. As with trajectory 2 evaluations, the results of the multiple model adaptive filter algorithm and the individual filters are presented. All of the trends observed in the trajectory 2 evaluations are repeated here. The sample mean errors tend to be higher than those for the trajectory 2 evaluations since the period over which temporal averaging is performed includes the transients created by the termination of the pull-up maneuver; thus the plots are particularly important here for appropriate insights.

Once again, the performance plots shown are only of the

TABLE V-6

TRAJECTORY 3 EVALUATIONS (2G)

CASE	$\bar{x}_e(-)$ / $\bar{\sigma x}_e(-)$	$\bar{x}_e(+)$ / $\bar{\sigma x}_e(+)$	$\bar{cx}_e(-)$ / $\bar{\sigma cx}_e(-)$	$\bar{cx}_e(+)$ / $\bar{\sigma cx}_e(+)$	$\bar{y}_e(-)$ / $\bar{\sigma y}_e(-)$	$\bar{y}_e(+)$ / $\bar{\sigma y}_e(+)$	$\bar{cy}_e(-)$ / $\bar{\sigma cy}_e(-)$	$\bar{cy}_e(+)$ / $\bar{\sigma cy}_e(+)$
L2T3MF	-.428/ .459	-.399/ .428	-.434/ .436	-.392/ .379	.049/ .588	.032/ .582	.027/ .586	.001/ .564
L2T3F1	-.323/ .314	-.260/ .273	-.236/ .259	-.105/ .122	.418/ .208	.304/ .182	.209/ .192	-.028/ .078
L2T3F2	-.012/ .337	.009/ .284	-.022/ .311	.009/ .223	.116/ .229	.095/ .266	.092/ .299	.061/ .235
G2T3MF	.084/ .120	.080/ .106	.062/ .143	.055/ .082	-.246/ .126	-.296/ .114	-.316/ .130	-.406/ .070
G2T3F1	.102/ .122	.098/ .109	.080/ .145	.073/ .085	-.230/ .127	-.28-/ .115	-.300/ .132	-.390/ .074
G2T3F2	.047/ .197	.045/ .159	.028/ .171	.025/ .084	.068/ .174	.047/ .139	.050/ .165	.022/ .077
C2T3MF	.332/ .126	.309/ .110	.285/ .149	.245/ .058	.043/ .133	-.018/ .122	-.041/ .144	-.146/ .099
C2T3F1	.349/ .129	.326/ .113	.302/ .151	.262/ .091	.075/ .136	.014/ .125	-.009/ .147	-.115/ .103
C2T3F2	.069/ .200	.060/ .160	.046/ .173	.340 .085	.078/ .178	.053/ .141	.059/ .167	.027/ .078

TABLE V-7

TRAJECTORY 3 EVALUATIONS (10G)

CASE	$\overline{x}_e(-)$ / $\overline{\sigma x}_e(-)$	$\overline{x}_e(+)$ / $\overline{\sigma x}_e(+)$	$\overline{cx}_e(-)$ / $\overline{\sigma cx}_e(-)$	$\overline{cx}_e(+)$ / $\overline{\sigma cx}_e(+)$	$\overline{y}_e(-)$ / $\overline{\sigma y}_e(-)$	$\overline{y}_e(+)$ / $\overline{\sigma y}_e(+)$	$\overline{cy}_e(-)$ / $\overline{\sigma cy}_e(-)$	$\overline{cy}_e(+)$ / $\overline{\sigma cy}_e(+)$
L10T3MF	-.469/ .415	-.430/ .383	-.473/ .384	-.419/ .326	.043/ .419	-.078/ .402	-.055/ .404	-.236/ .362
L10T3F2	-.088/ .298	-.053/ .249	-.091/ .272	-.039/ .180	.337/ .281	.219/ .244	.248/ .278	.076/ .196
G10T3MF	.037/ .197	.045/ .161	.022/ .170	.034/ .084	.145/ .171	.048/ .137	-.088/ .161	-.043/ .073
G10T3F2	.038/ .197	.047/ .159	.024/ .171	.035/ .083	.185/ .172	.089/ .138	.129/ .163	-.001/ .075
C10T3MF	.037/ .200	.044/ .161	.022/ .172	.031/ .084	.216/ .175	.096/ .138	.146/ .163	-.013/ .073
C10T3F2	.044/ .201	.052/ .161	.029/ .173	.039/ .084	.237/ .176	.118/ .140	.168/ .166	.012/ .075

TABLE V-8

TRAJECTORY 3 EVALUATIONS (20G)

CASE	$\overline{x}_e(-)$ / $\overline{\sigma x}_e(-)$	$\overline{x}_e(+)$ / $\overline{\sigma x}_e(+)$	$\overline{cx}_e(-)$ / $\overline{\sigma cx}_e(-)$	$\overline{cx}_e(+)$ / $\overline{\sigma cx}_e(+)$	$\overline{y}_e(-)$ / $\overline{\sigma y}_e(-)$	$\overline{y}_e(+)$ / $\overline{\sigma y}_e(+)$	$\overline{cy}_e(-)$ / $\overline{\sigma cy}_e(-)$	$\overline{cy}_e(+)$ / $\overline{\sigma cy}_e(+)$
L20T3MF	-.508/ .419	-.432/ .388	-.485/ .397	-.375/ .333	.724/ .352	.488/ .328	.538/ .345	.188/ .289
L20T3F2	-.100/ .280	-.025/ .227	-.078/ .257	.032/ .159	.807/ .268	.576/ .228	.639/ .277	.301/ .184
G20T3MF	-.037/ .200	.006/ .164	-.034/ .171	.024/ .084	.094/ .166	-.093/ .135	-.012/ .155	-.265/ .069
G20T3F2	-.033/ .200	.009/ .164	-.031/ .172	.026/ .084	.080/ .177	-.106/ .148	-.025/ .162	-.275/ .082
C20T3MF	-.083/ .203	-.025/ .165	-.073/ .173	.002/ .084	.332/ .177	.101/ .143	.204/ .168	-.101/ .087
C20T3F2	-.071/ .202	-.014/ .164	-.060/ .173	.014/ .083	.384/ .196	.154/ .165	.255/ .182	-.048/ .105

y-dynamics mean errors since that is the direction of the target maneuver. It is easy to see the termination of the maneuver at t=3.5 seconds. In the 2g case, the heavy weighting of the small field of view filter is still evident in comparisons of the figures of merit and the plots for the multiple model filter and the small field of view filter. The performance of the multiple model filter appears to be much worse than the performance of either of the individual filters in the Millner tracker formulation. The plots of this case, Figure A-11, show that while the times of the pull-up and continuation maneuver are clearly evident, the standard deviations become very large, especially after termination of the maneuver. During this time period, the multiple filter is having a difficult time selecting the correct filter model. The observed target behavior is that of a constant velocity target buth the approach of the minimum range/maximum passing rate condition indicates that the field of view should be larger to handle a more dynamic trajectory. No such behavior is noted in the plots for the extended Kalman filter tracker, where again the poor tuning of the small field of view filter may be hindering the filter from adaptively reducing the size of the effective tracking window. As before, the extended Kalman filter tracker, using the Gauss-Markov model is showing much larger biases. Finally, as with trajectory 2 evaluations, the high setting of the strength of the target acceleration white noise tends to mask out low g maneuvers.

5.5.4 Evaluations Using Trajectory 4. Table V
sents the results of the simulations for trajectory
gets performing 2, 10, and 20g constant speed turns
ever, because early results indicated that perform
the trackers did not vary greatly from the perf
observed in trajectory 2 evaluations, simulations we
performed for the linear Kalman filter/correlator t
The reason for the similarity is that, even with
target, the out-of-plane component of the target path
to remain very small due to the short duration of th
lation. At the same time, because the target turns
the tracker during this maneuver, the target intensit
function changes considerably over the simulatio
This demonstrates the tracker's ability to mainta
estimates of a time varying shape function, and th
tracking performance.

The performance plots of these cases are Figu
thru A-24. The plots of the x-dynamics mean err
included for the 2g case to demonstrate that the majo
the maneuver still takes place in the y- directio
plots of the y-dynamics error statistics can be comp
those for the trajectory 2 evaluation (Figures A-4
10) to see the similarity of tracker performance for
types of trajectories.

5.5.5 Changes in Signal-to-Noise Ratio. As
earlier in this chapter, the nominal SNR for all simu
was set 20. To test the multiple filter's robustne

TABLE V-9

TRAJECTORY 4 EVALUATIONS

CASE	$\overline{x}_e(-)$ / $\overline{\sigma x}_e(-)$	$\overline{x}_e(+)$ / $\overline{\sigma x}_e(+)$	$\overline{cx}_e(-)$ / $\overline{\sigma cx}_e(-)$	$\overline{cx}_e(+)$ / $\overline{\sigma cx}_e(+)$	$\overline{y}_e(-)$ / $\overline{\sigma y}_e(-)$	$\overline{y}_e(+)$ / $\overline{\sigma y}_e(+)$	$\overline{cy}_e(-)$ / $\overline{\sigma cy}_e(-)$	$\overline{cy}_e(+)$ / $\overline{\sigma cy}_e(+)$
L2T4MF	.208/ .191	.183/ .172	.142/ .195	.090/ .087	-.172/ .343	-.132/ .332	-.119/ .345	-.040/ .301
L2T4F1	.226/ .174	.194/ .159	.145/ .184	.079/ .057	-.182/ .166	-.142/ .152	-.118/ .181	-.033/ .069
L2T4F2	.202/ .301	.197/ .250	.174/ .302	.166/ .214	.120/ .323	.154/ .280	.137/ .330	.187/ .253
L10T4MF	.369/ .459	.307/ .433	.271/ .460	.165/ .408	-.141/ .730	.019/ .719	.004/ .695	.257/ .656
L10T4F2	.154/ .295	.106/ .249	.092/ .290	.022/ .204	.345/ .286	.501/ .254	.458/ .280	.687/ .206
L20T4MF	.386/ .418	.202/ .395	.214/ .428	-.057/ .381	.022/ .526	.201/ .514	.159/ .518	.419/ .488
L20T4F2	.359/ .264	.176/ .220	.193/ .264	-.073/ .159	.014/ .256	.195/ .222	.152/ .252	.416/ .162

change in the expected SNR, a few test cases were performed using the Millner tracker formulation at a SNR=10. No retuning of the filter was done for this lower SNR. This value also corresponds to a reasonable SNR for a tracking scenario (3). The results are presented in Table V-10 for a trajectory 2 target performing 2, 10, and 20g pull-up maneuvers. The associated plots are Figures A-25 thru A-27. As with trajecotry 4 evaluations, these plots can be compared to the original plots for the trajectory 2 evaluations (Figures A-4 thru A-10), to see the similarity in performance. While there is some increase in the tracker sample rms errors, it still does a reasonable job of tracking even the highly dynamic 20g target. Attempts to reduce the SNR further to a value of unity resulted in completely divergent filter behavior, as seen previously in the study by Harnly and Jenson (3) for single hot-spot targets.

5.5.6 Ad Hoc Changes to Filter Conditional Probabilities. The final test performed during this research was done to establish some upper bound on the performance of the multiple model filter's ability to switch from one filter model to the other when a step change occurs in the truth model description of the target. This was accomplished by artificially providing the filter with perfect knowledge of when a maneuver was initiated. The result is a step change in the conditional probabilities of the single filters to correspond with the step change in the truth model. Because the filter transients have already died out by the time

temporal averaging of the statistics takes place, this ad hoc change in the conditional probabilities had little or no effect on the figures of merit. However, the plots (Figures A-28 and A-29; B-18 and B-19; and C-18 and C-19) show that, as compared to the trajectory 2 evaluation plots, there was little effect on the time it took the filter transients to die out after the maneuver was initiated. What was changed, was the mean peak error of each tracker type, which was reduced by a factory of two. Since this was the only significant deviation in the performances of the multiple filter and the ad hoc cases, it was decided that the multiple filter performs reasonably well as compared to this theoretical upper bound.

TABLE V-10

TRAJECTORY 2 EVALUATIONS (SNR=10)

CASE	$\overline{x}_e(-)$ / $\overline{\sigma x}_e(-)$	$\overline{x}_e(+)$ / $\overline{\sigma x}_e(+)$	$\overline{cx}_e(-)$ / $\overline{\sigma cx}_e(-)$	$\overline{cx}_e(+)$ / $\overline{\sigma cx}_e(+)$	$\overline{y}_e(-)$ / $\overline{\sigma y}_e(-)$	$\overline{y}_e(+)$ / $\overline{\sigma y}_e(+)$	$\overline{cy}_e(-)$ / $\overline{\sigma cy}_e(-)$	$\overline{cy}_e(+)$ / $\overline{\sigma cy}_e(+)$
LS2T2MF	.256/ .182	.231/ .162	.187/ .197	.135/ .085	-.139/ .331	-.098/ .314	-.085/ .334	-.004/ .283
LS10T2MF	.312/ .378	.244/ .353	.228/ .379	.125/ .329	.051/ .766	.224/ .760	.184/ .765	.440/ .747
LS20T2MF	.595/ .309	.364/ .272	.390/ .316	.049/ .237	-.223/ .423	.075/ .407	.003/ .428	.441/ .393

VI. Conclusions and Recommendations

The multiple model filter algorithm has been shown to perform well against a wide dynamic range of targets. While not capable of performing as well as the original tracker formulations against relatively benign target trajectories, it does offer the advantage of being able to maintain lock onto highly dynamic targets that the original trackers were unable to follow. This increase in tracker capability would seem to offset the additional memory storage requirements and computational burden imposed by the multiple model filter algorithm. The additional time required to implement the multiple model filter can be tempered by using parallel processing techniques to process each individual filter, instead of processing them sequentially, as was done in this research.

From an implementation standpoint, it is easier to incorporate the multiple model algorithm into the linear Kalman filter/correlator tracker formulation than into the extended Kalman filter tracker. The linearity of the filter as well as the low dimensionality of the filter measurement vector for the Millner tracker formulation made it possible to implement the multiple model algorithm without any of the approximations and ad hoc changes necessary to implement the same algorithm in the extended Kalman filter tracker. The high dimensionality of the filter measurement vector for the extended Kalman filter tracker also drastically increased the memory storage requirements and computational loading of

the multiple model filter. Using identical FLIR measurement algorithms for the computer simulations of both tracker types, the simulation for the multiple model filter for the extended Kalman filter tracker occupied twice as much memory space and took approximately 30 percent more time to execute than the same simulation incorporating the linear Kalman filter/correlator structure. For this feasibility study, no attempts were made to minimize the memory requirements of each simulation other than what would be the result of structured programming techniques. Therefore, the memory required to run either simulation is subject to change and the differences between the simulations may also change when more efficient programming techniques are applied.

The results of the computer simulations show that the extended Kalman filter tracker significantly outperforms the Millner tracker formulation when evaluating tracker performance against highly dynamic targets. Furthermore, the constant turn rate acceleration model is superior to the Gauss-Markov model as a model of true target behavior. Even at very benign trajectories, where the poor tuning of the small field of view filter was a problem, the Kozemchak tracker formulation possessed much smaller sample standard deviations of tracker errors. For a laser weapon system, this may be more desirable than the lower sample mean errors achieved by the Millner tracker formulation, as long as the rms errors of both tracker types are about the same. The filter transients and mean peak errors were also smaller for

the extended Kalman filter tracker. Overall, for this configuration of the multiple model adaptive algorithm, with two independent Kalman filters, the improved performance of the extended Kalman filter tracker is worth the additional memory storage and computational burden.

Recommendations

Further study is recommended in order to investigate problems encountered or tasks not accomplished in this research. These areas include:

* Improved filter tuning of the small field of view filter for the extended Kalman filter tracker

* Addition of more filters in the bank of independent filters, to imclude additional target dynamics levels, different fields of view, etc.

* Implementation of the constant turn rate target acceleration model in the linear Kalman filter/correlator tracker

* Realistic changes to the truth model target dynamics. A more realistic model than the step change of the truth model at the inititation of a maneuver may show that the dynamic range achieved by the multiple model filter is not necessary for realistic tracking scenarios.

* Test trackers against targets whose shape functions are rapidly changing. This includes targets performing roll maneuvers as well as non-realistic scenarios where the size, shape, intensity, and number of hot-spots is changing.

* Improved ways of handling the approximation made to implement the multiple model algorithm in the extended Kalman filter tracker

* Extending the length of the simulation to investigate the minimum range/maximum passing rate condition and the response of the multiple model adaptive filter to it

* Determine how changes in the sampling rate of the simulation affects tracker performance

* Determine the algorithm's sensitivity to different settings for the standard deviation of atmospheric jitter

* Determine the algorithm's sensitivity to the value of the lower bound on conditional probabilities (requiring filter retuning for best performance)

* Other robustness studies should be performed to determine the algorithm's sensitivity to target shape, separation of the hot-spots, etc.

Bibliography

1. Bedworth, David D. Industrial Systems Planning, Analysis, Control. New York: The Ronald Press Company, 1973.

2. Flynn, Patrick M., "Alternative Dynamics Models and Multiple Model Filtering for a Short Range Tracker," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1981.

3. Harnly, Douglas A. and Robert L. Jensen, "An Adaptive Distributed-Measurement Extended Kalman Filter for a Short Range Tracker," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1979.

4. Hogge, C.B. and R.R. Butts, "Frequency Spectra for the Geometric Representation of Wavefront Distortions Due to Atmospheric Turbulence," IEEE Transactions on Antennas and Propagation, Vol. AP-24, No. 2, March 1976.

5. Kozemchak, Mark R., "Enhanced Image Tracking: Analysis of Two Acceleration Models in Tracking Multiple Hot-Spot Images," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1982.

6. Maybeck, Peter S. Stochastic Models, Estimation and Control Volume I. New York: Academic Press Incorporated, 1979.

7. Maybeck, Peter S. Stochastic Models, Estimation and Control Volume II. New York: Academic Press Incorporated, 1982.

8. Maybeck, Peter S., Douglas A. Harnly and Robert L. Jensen, "Robustness of a New Infrared Target Tracker," Proceedings of IEEE National Aerospace and Electronics Conference, Dayton, Ohio, pp. 639-644, May 1980.

9. Maybeck, Peter S. and Daniel E. Mercier, "A Target Tracker Using Spatially Distributed Infrared Measurements," IEEE Transactions on Automatic Control, Vol. AC-25, No.2, pp. 222-225, April 1980.

10. Maybeck, Peter S. and Steven K. Rogers, "Adaptive Tracking of Dynamic Multiple Hot-Spot Target IR Images," Proceedings of IEEE Conference on Decision and Control, Orlando, Florida, pp. 1145-1151, December 1982.

11. Maybeck, Peter S., William H. Worsley and Patrick M. Flynn, "Investigation of Constant Turn-Rate Dynamics Models in Filters for Airborne Vehicle Tracking," Proceedings of IEEE National Aerospace and Electronics Conference, Dayton, Ohio, pp. 896-903, May 1982.

12. Mercier, Daniel E., "An Extended Kalman Filter for Use in a Shared Aperture Medium Range Tracker," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1978.

13. Millner, Paul P., "Enhanced Tracking of Airborne Targets Using a Correlator/Kalman Filter," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1982.

14. Rogers, Steven K., "Enhanced Tracking of Airborne Targets Using Forward Looking Infrared Measurements," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1981.

15. Singletery, James Jr., "Adaptive Laser Pointing and Tracking Problem," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1980.

16. The Analytic Sciences Corporation. Advanced Adaptive Optics Control Techniques. TR-966-1. Prepared for the Air Force Weapons Laboratory, Kirtland AFB, New Mexico, January 6, 1978.

17. Worsley, William H., "Comparison of Three Extended Kalman Filters for Air-to-Air Tracking," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1980.

Appendix A

This appendix contains the performance plots for the linear Kalman filter/enhanced correlator tracker configuration using a Gauss-Markov target acceleration model. These cases are identified according to the mnemonic code described in Figure V-3. The values for the parameters of these simulations are also found in Chapter V.

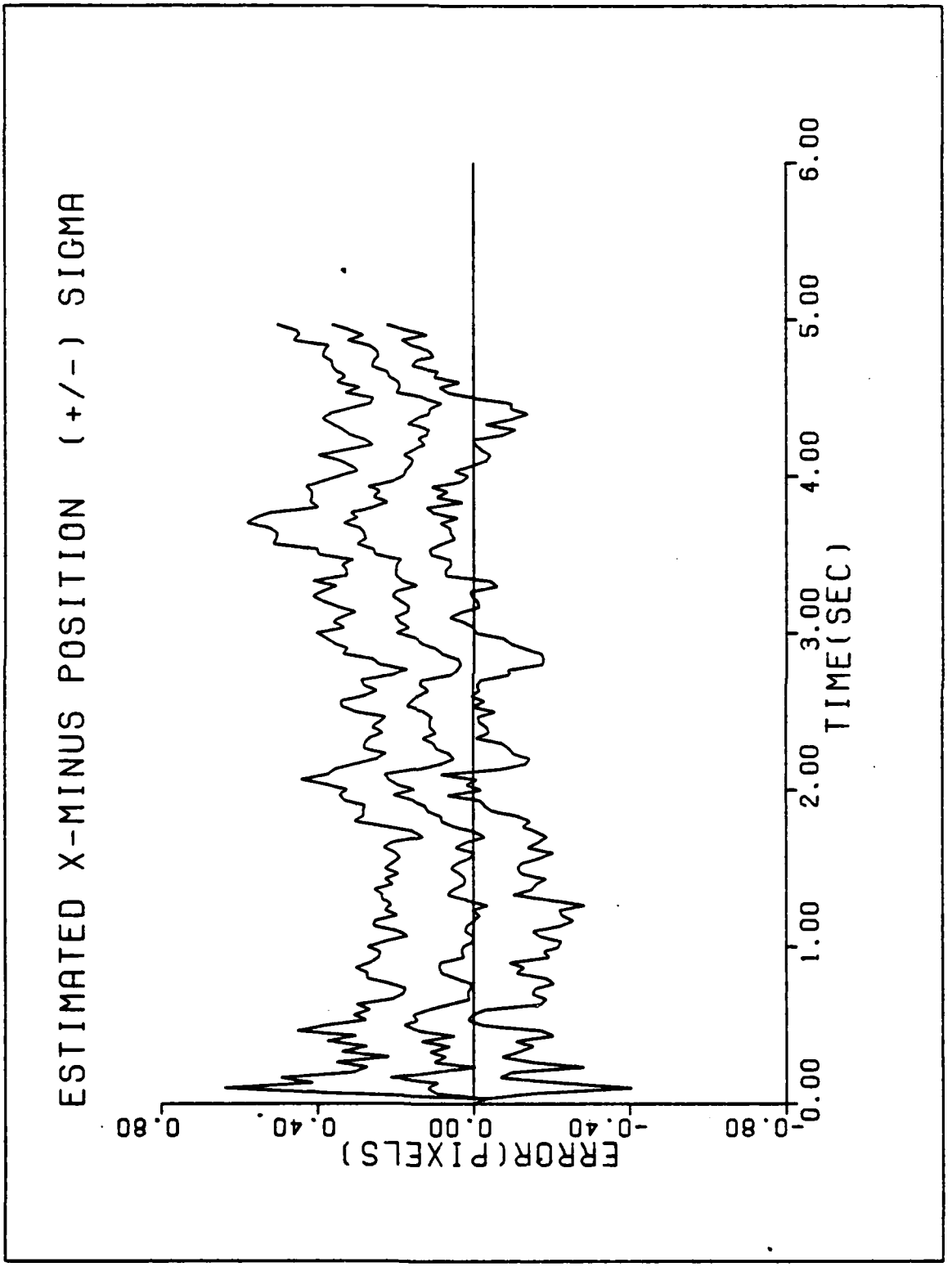


Figure A-1a. Performance Plot for LT1MF

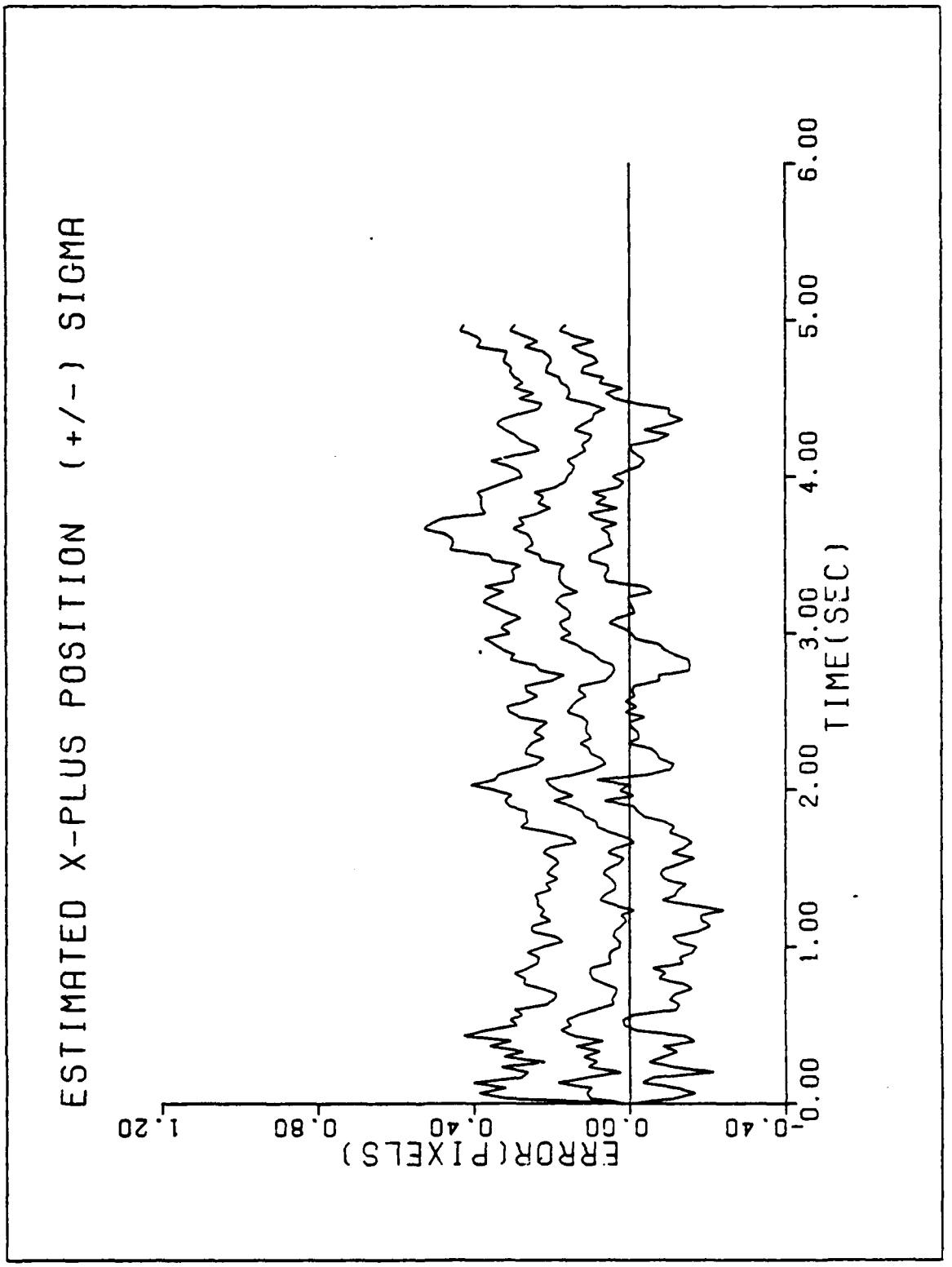


Figure A-1b. Performance Plot for LT1MF

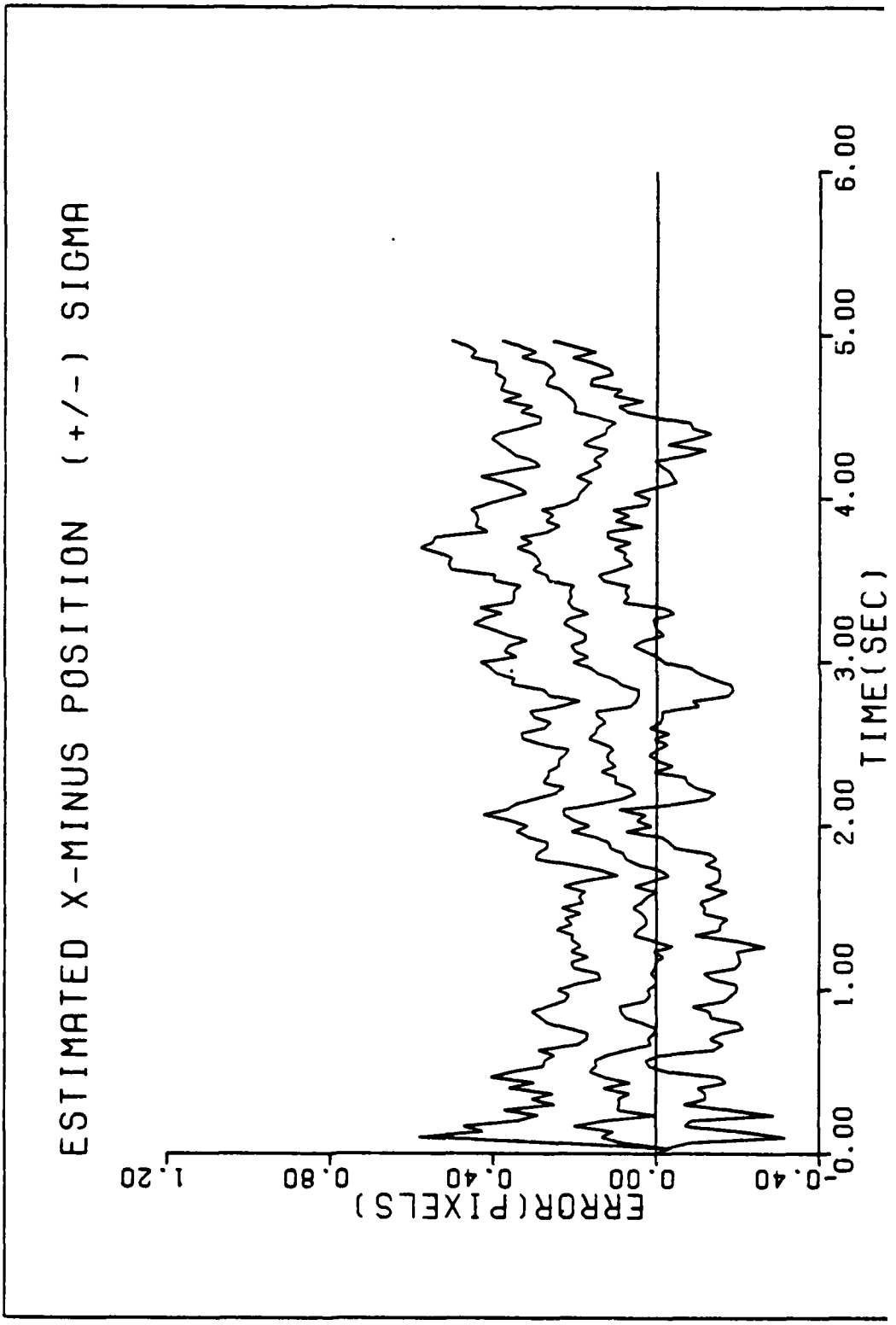


Figure A-2a. Performance Plot for LT1F1

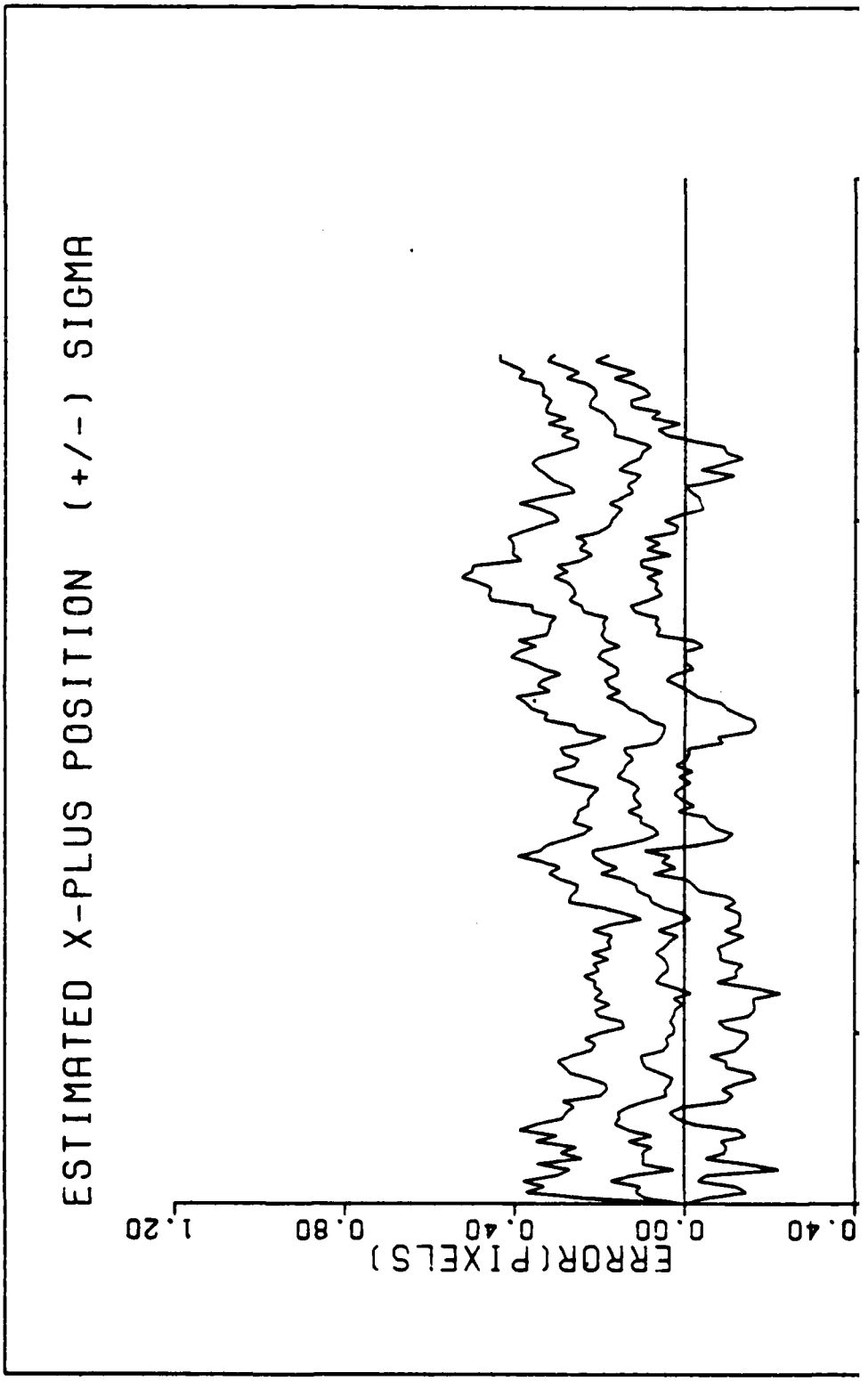


Figure A-2b. Performance Plot for LT1F1

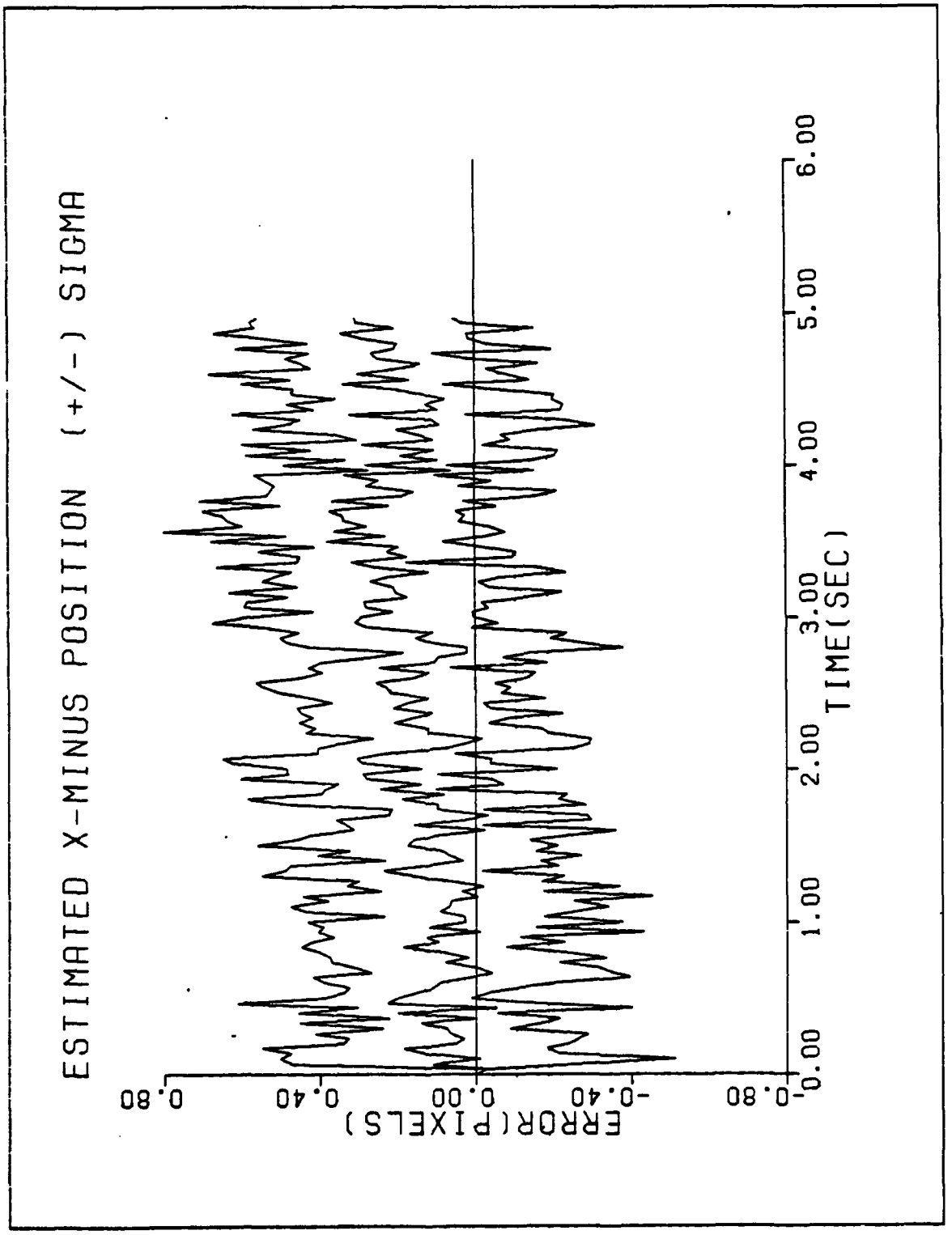


Figure A-3a. Performance Plot for LT1F2

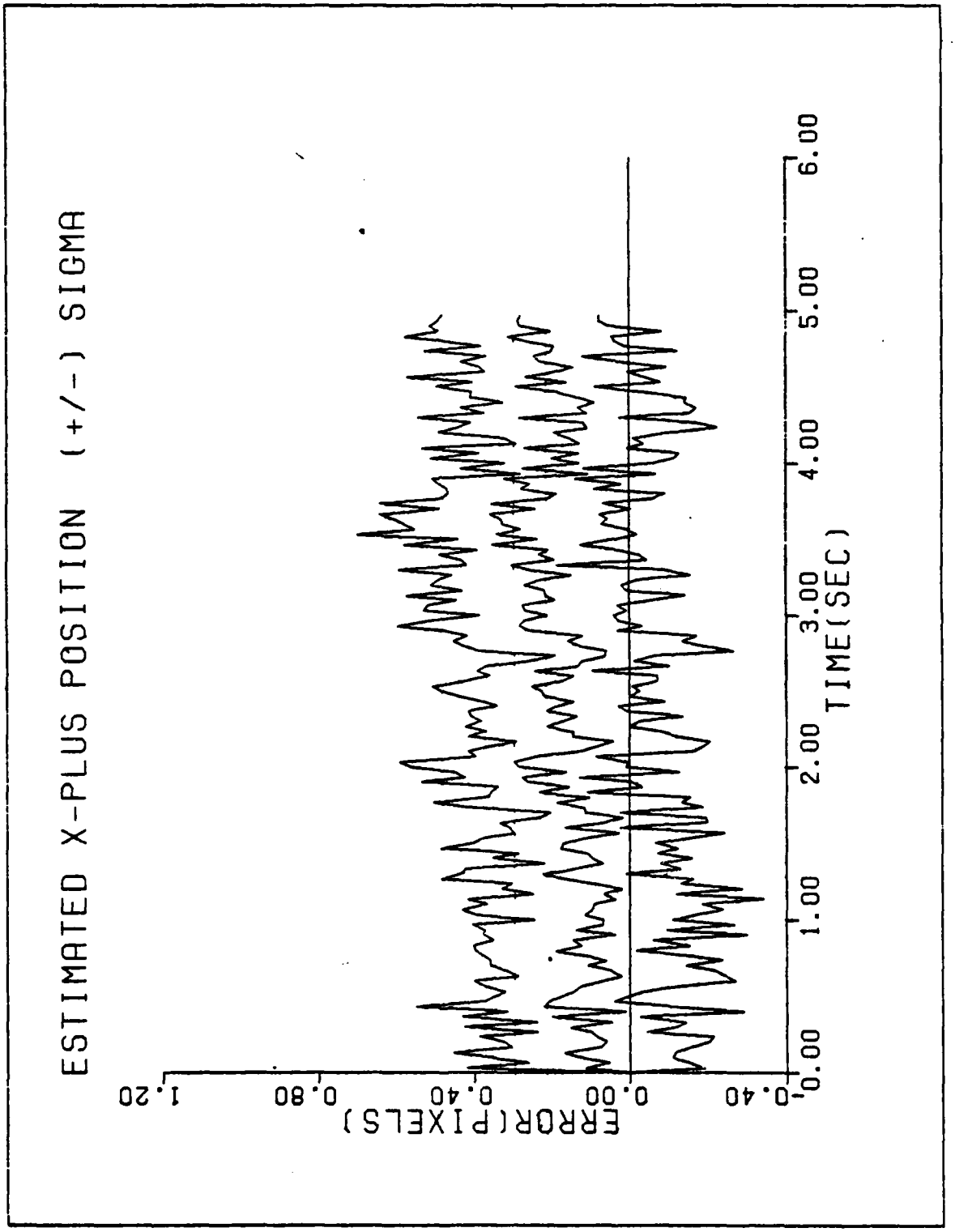


Figure A-3b. Performance Plot for LT1F2

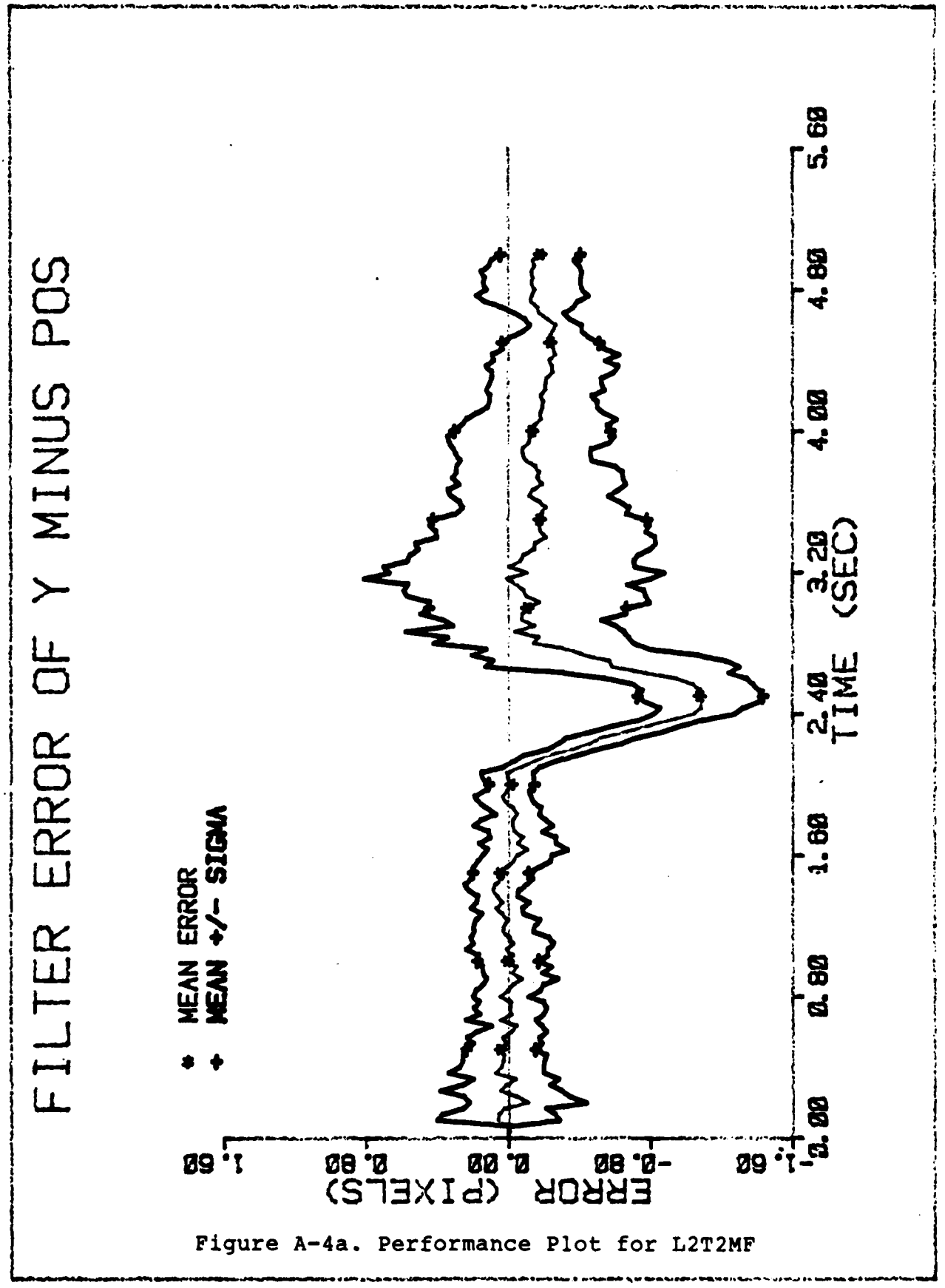


Figure A-4a. Performance Plot for L2T2MF

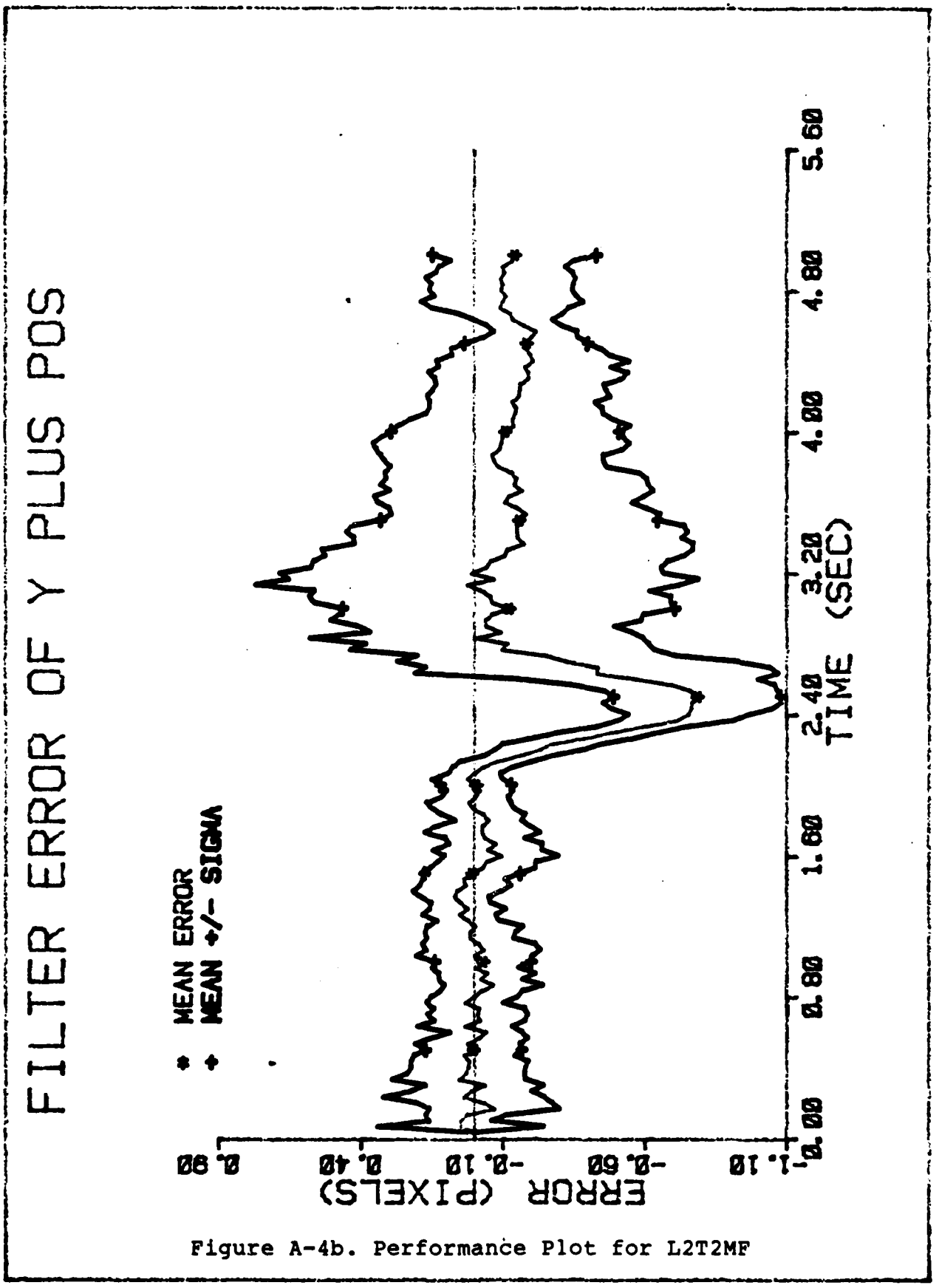


Figure A-4b. Performance Plot for L2T2MF

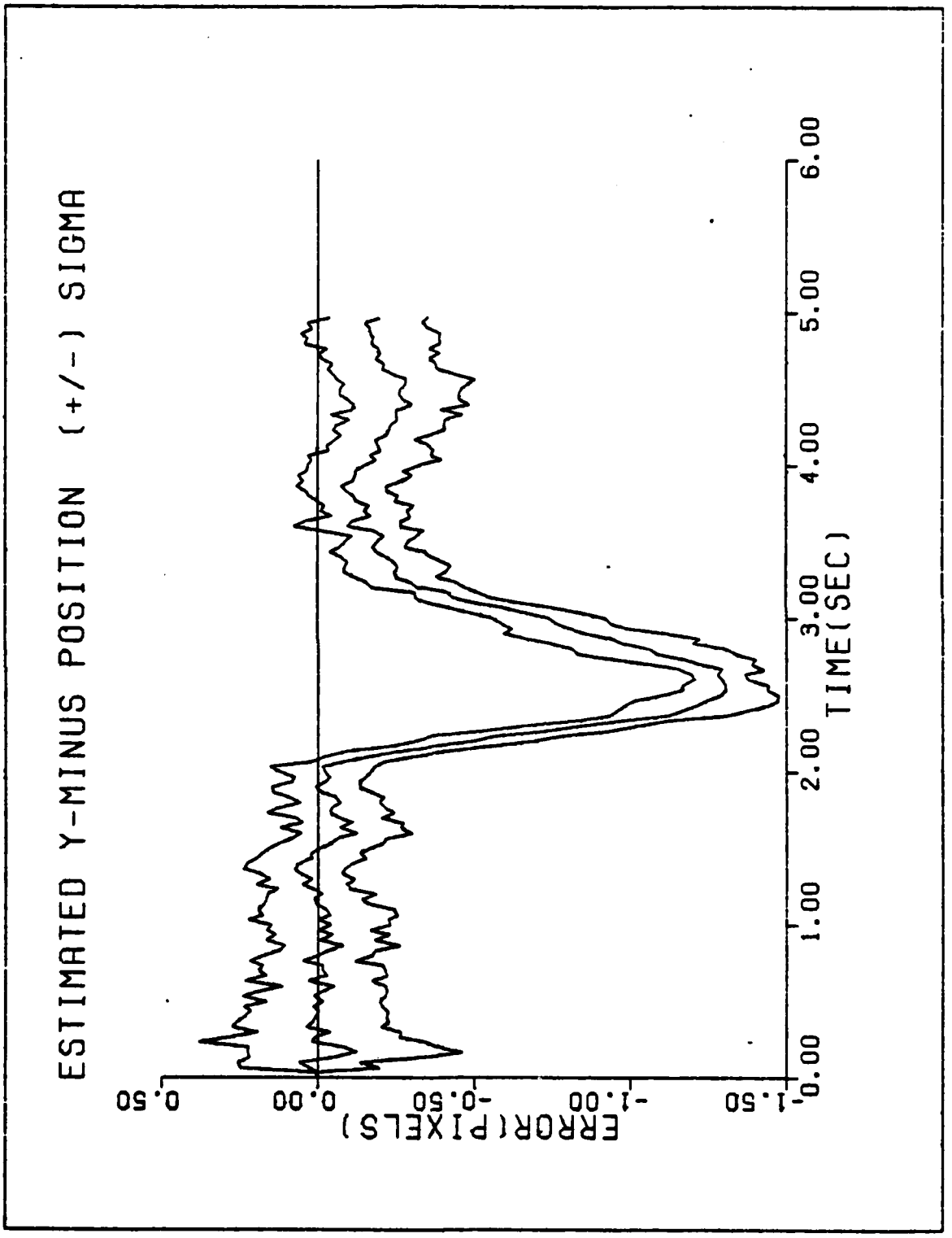


Figure A-5a. Performance Plot for L2T2F1

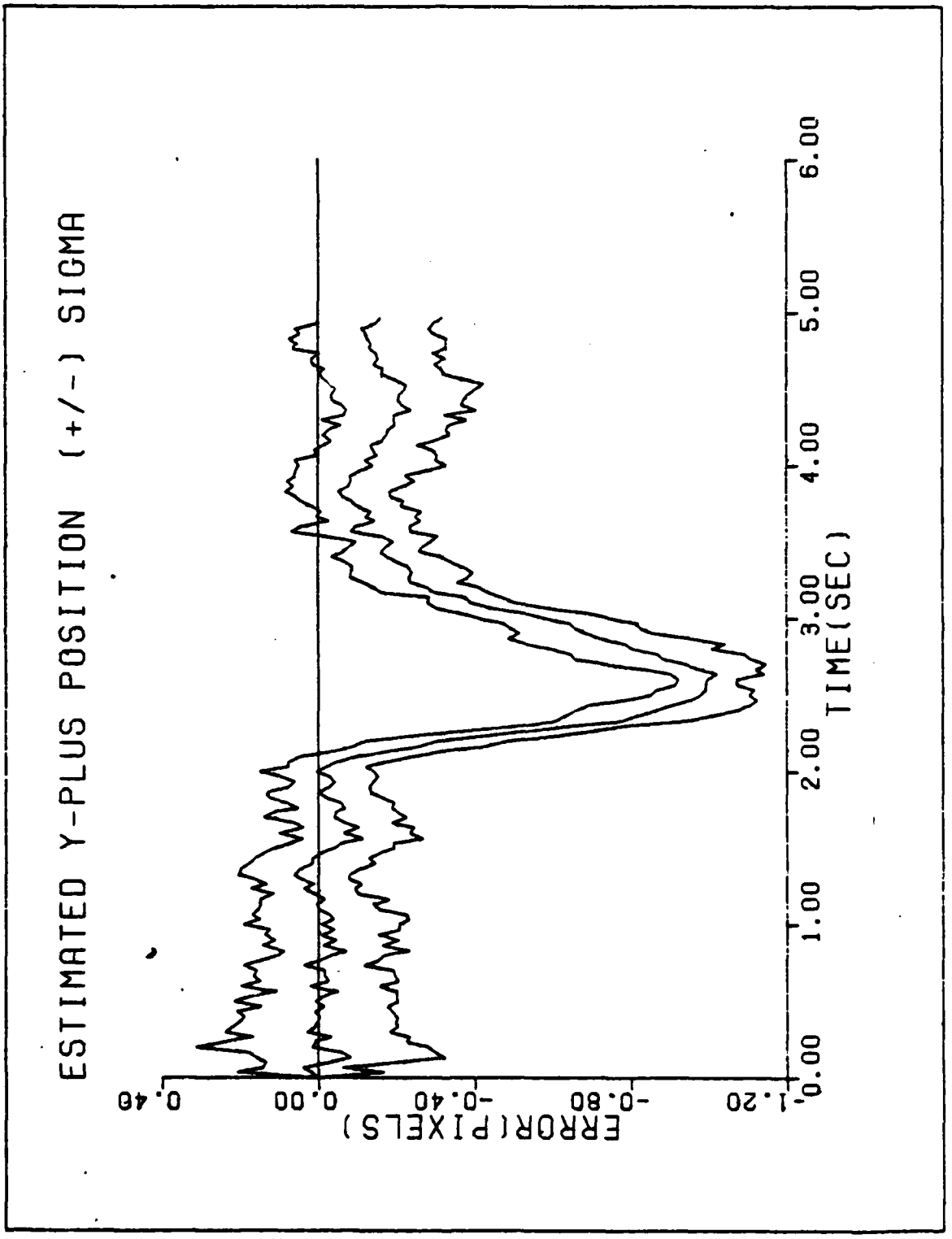


Figure A-5b. Performance Plot for L2T2F1

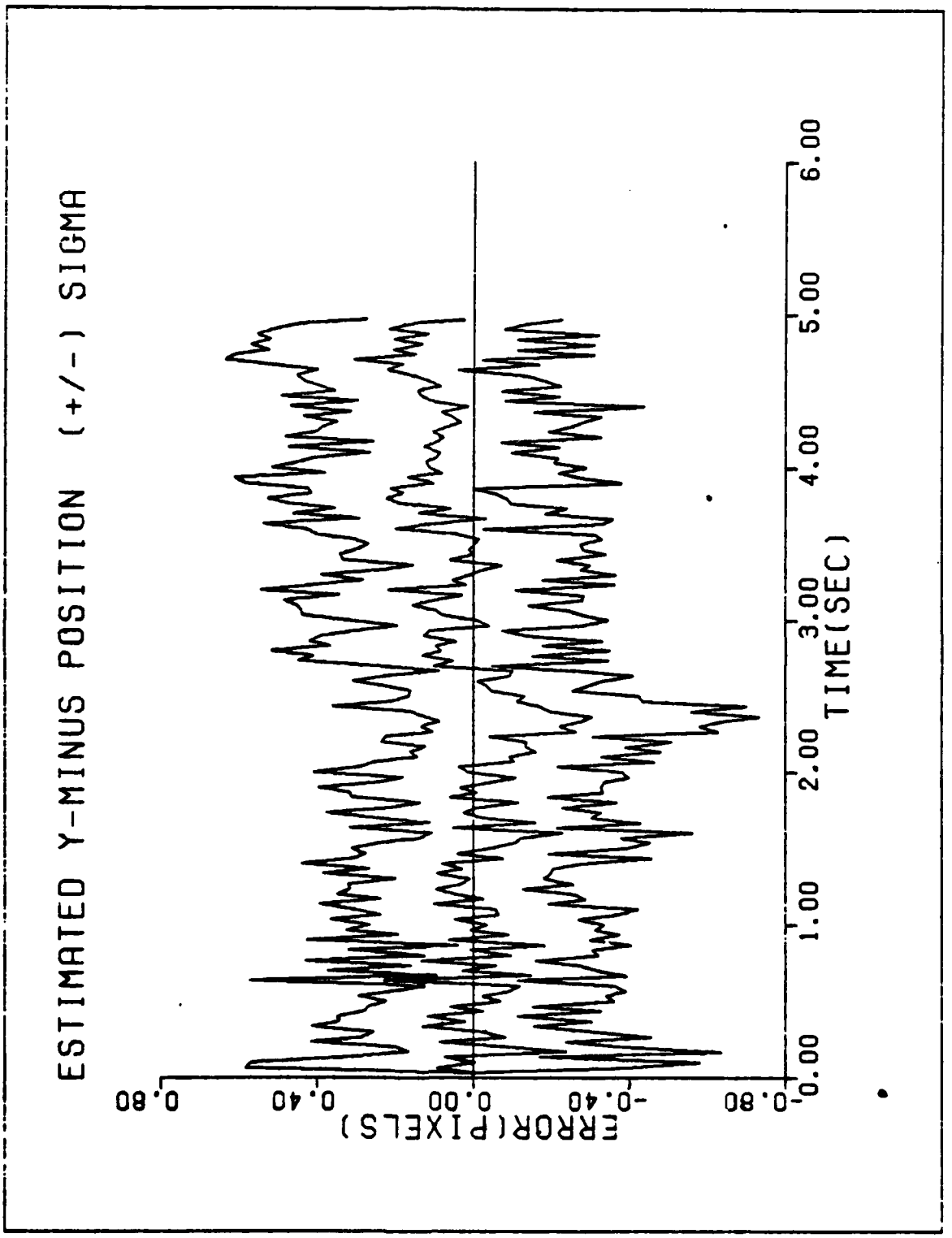


Figure A-6a. Performance Plot for L2T2F2

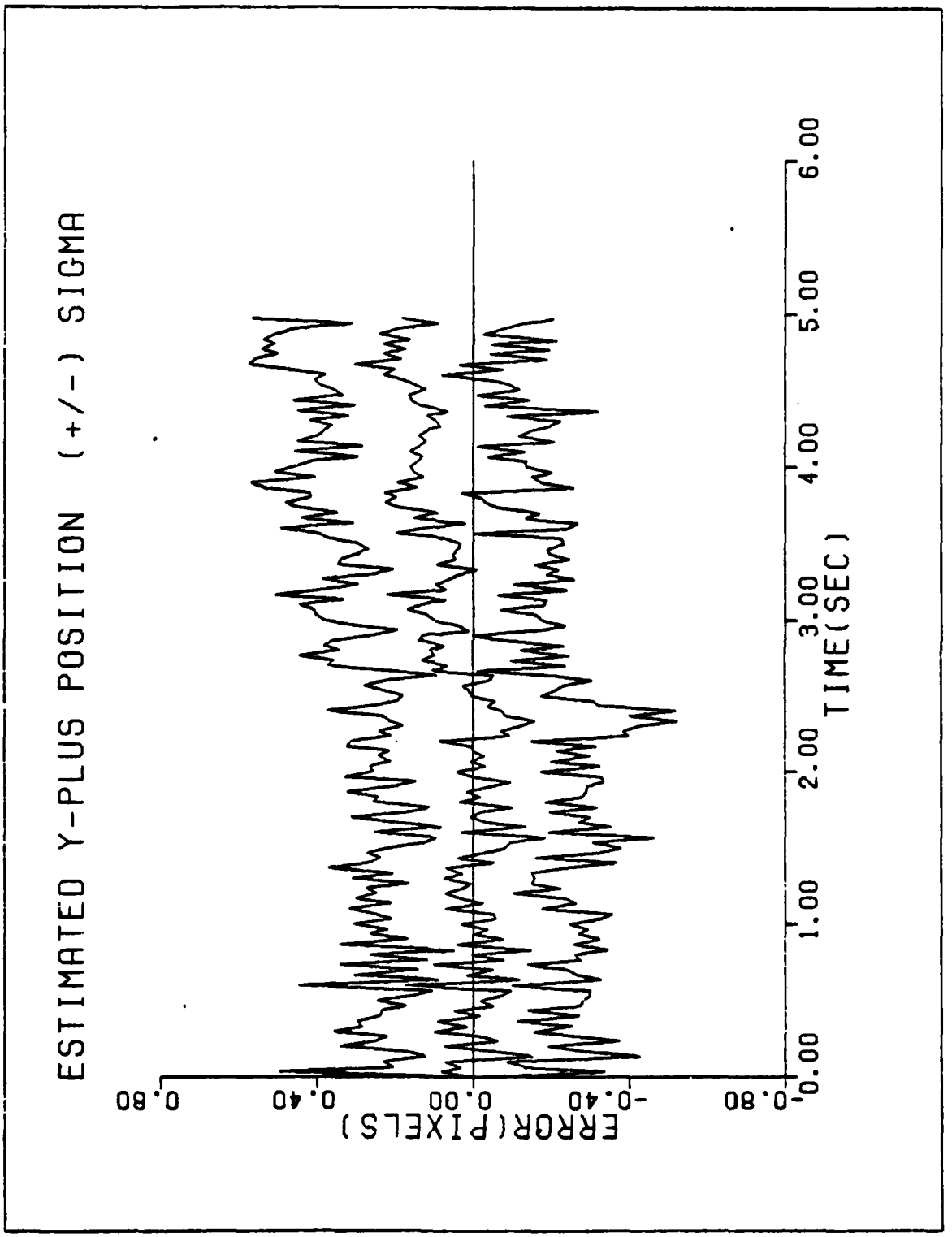


Figure A-6b. Performance Plot for L2T2F2

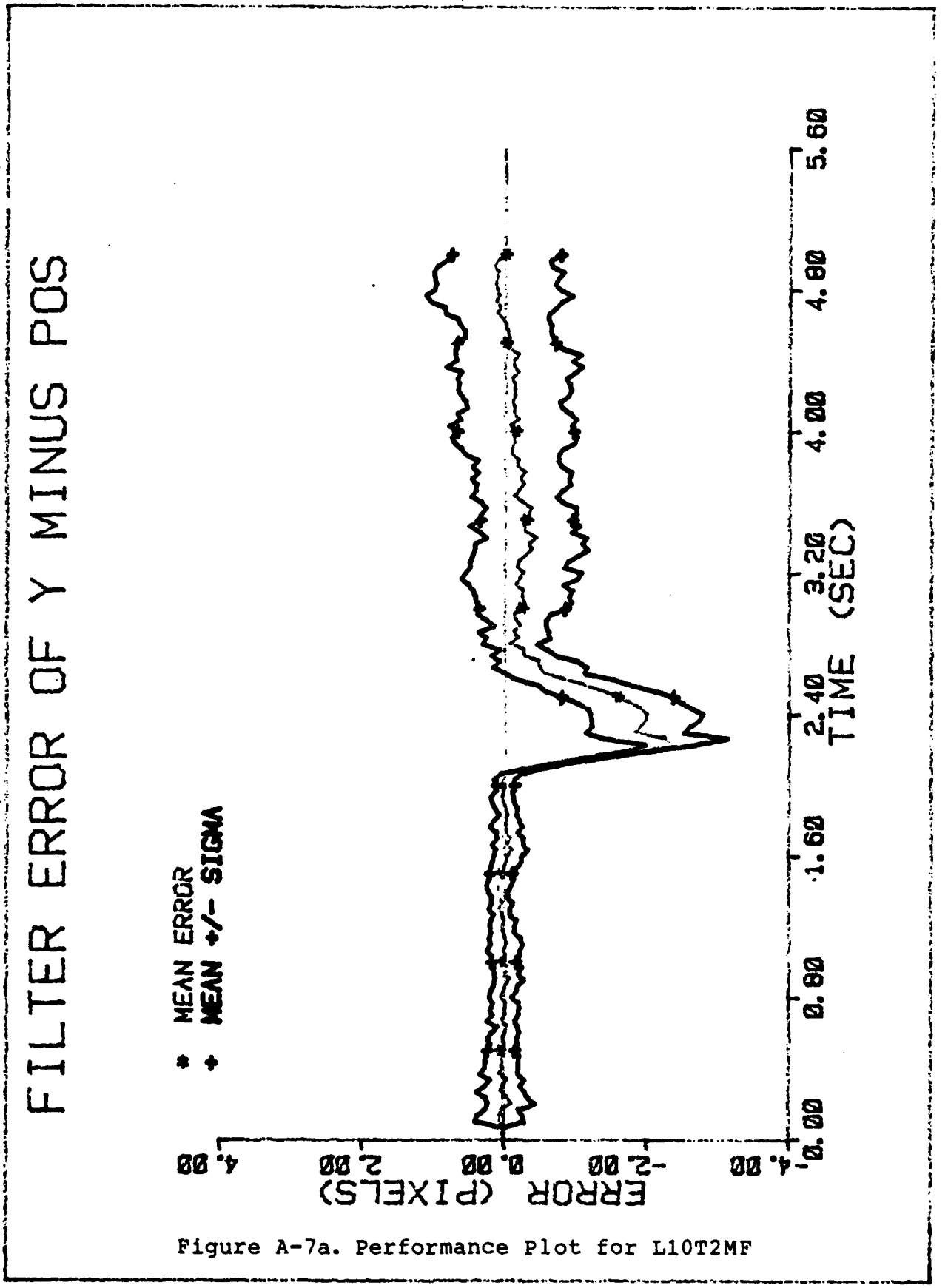


Figure A-7a. Performance Plot for L10T2MF

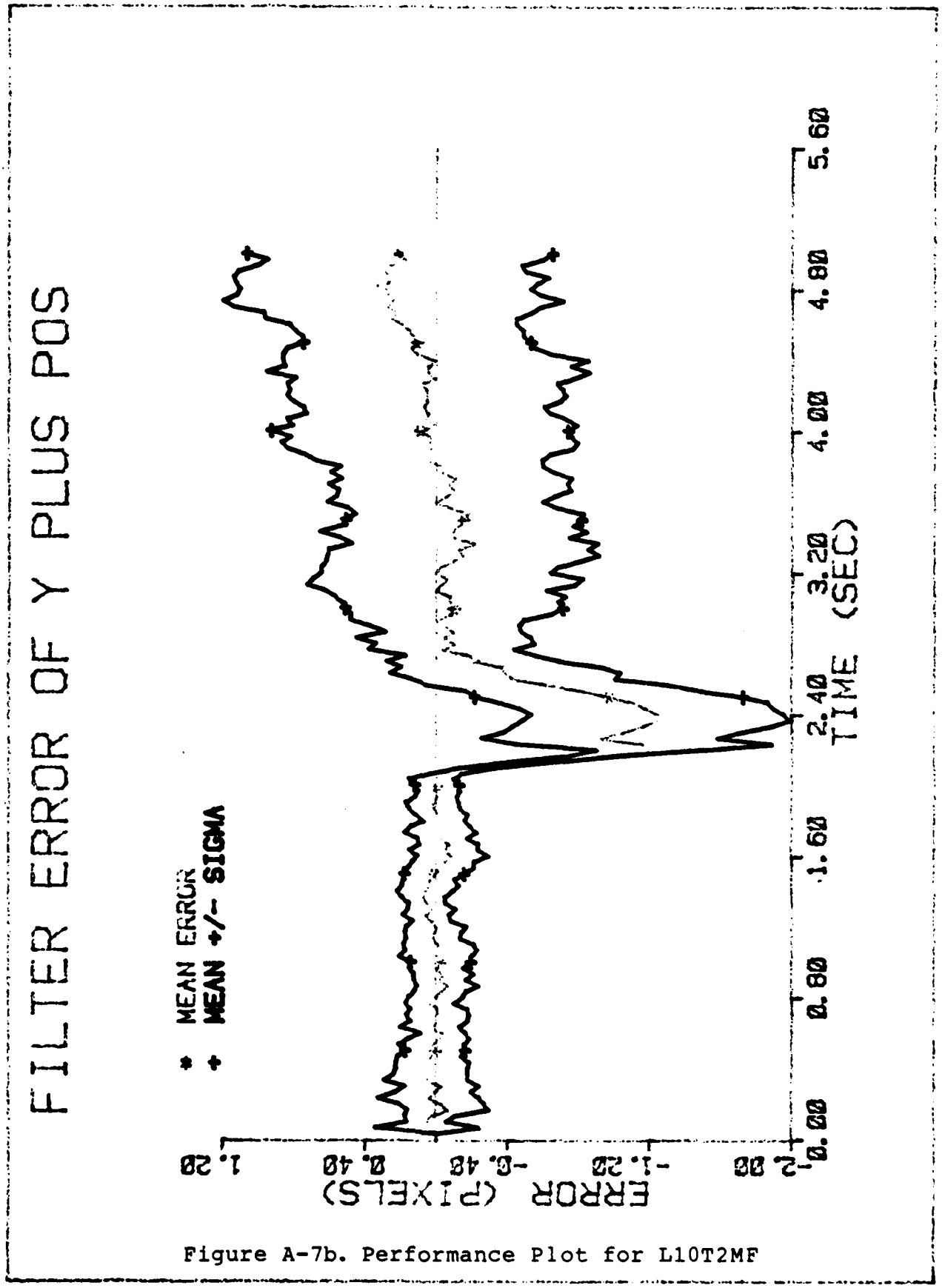


Figure A-7b. Performance Plot for L10T2MF

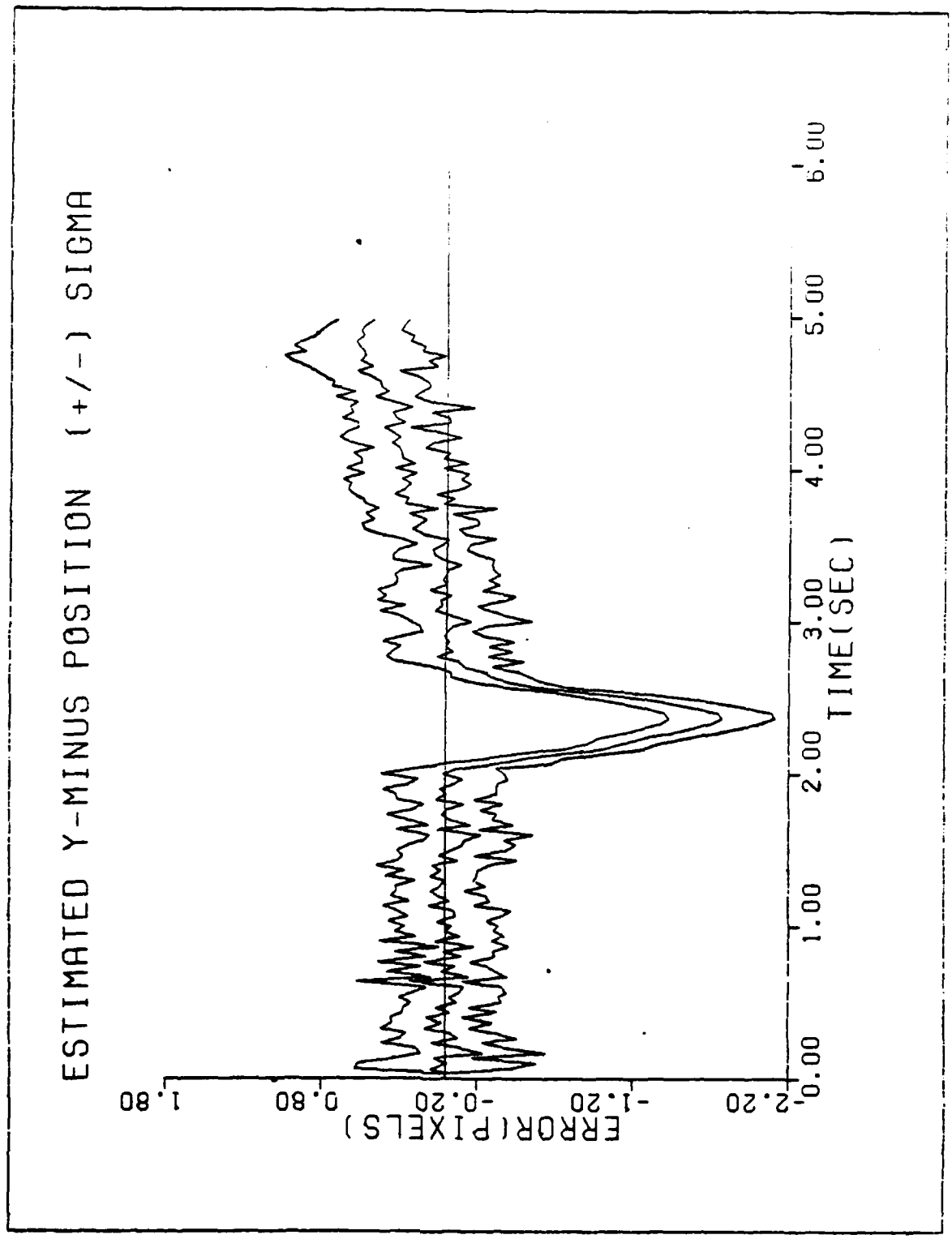


Figure A-8a. Performance Plot for L10T2F2

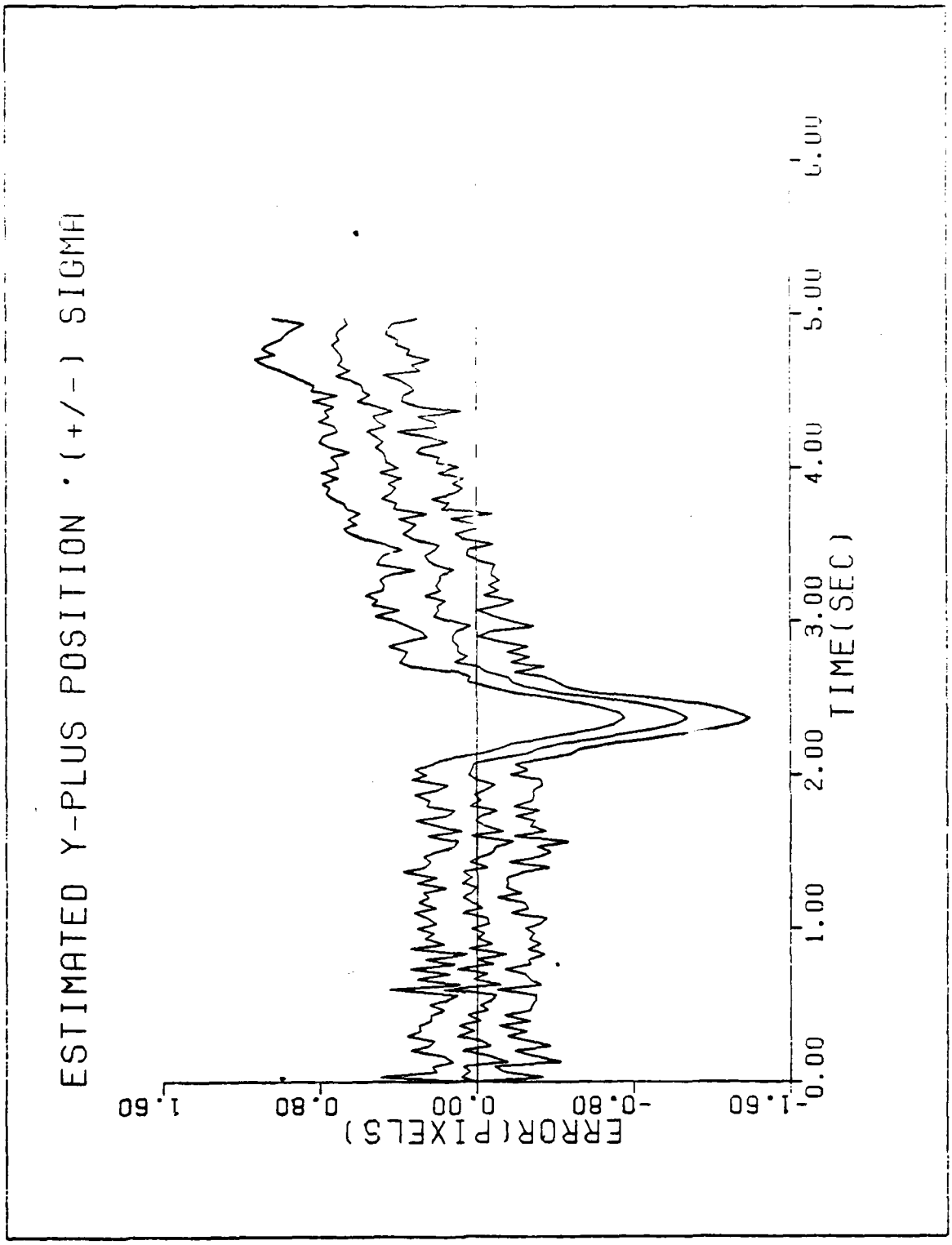


Figure A-8b. Performance Plot for L10T2F2

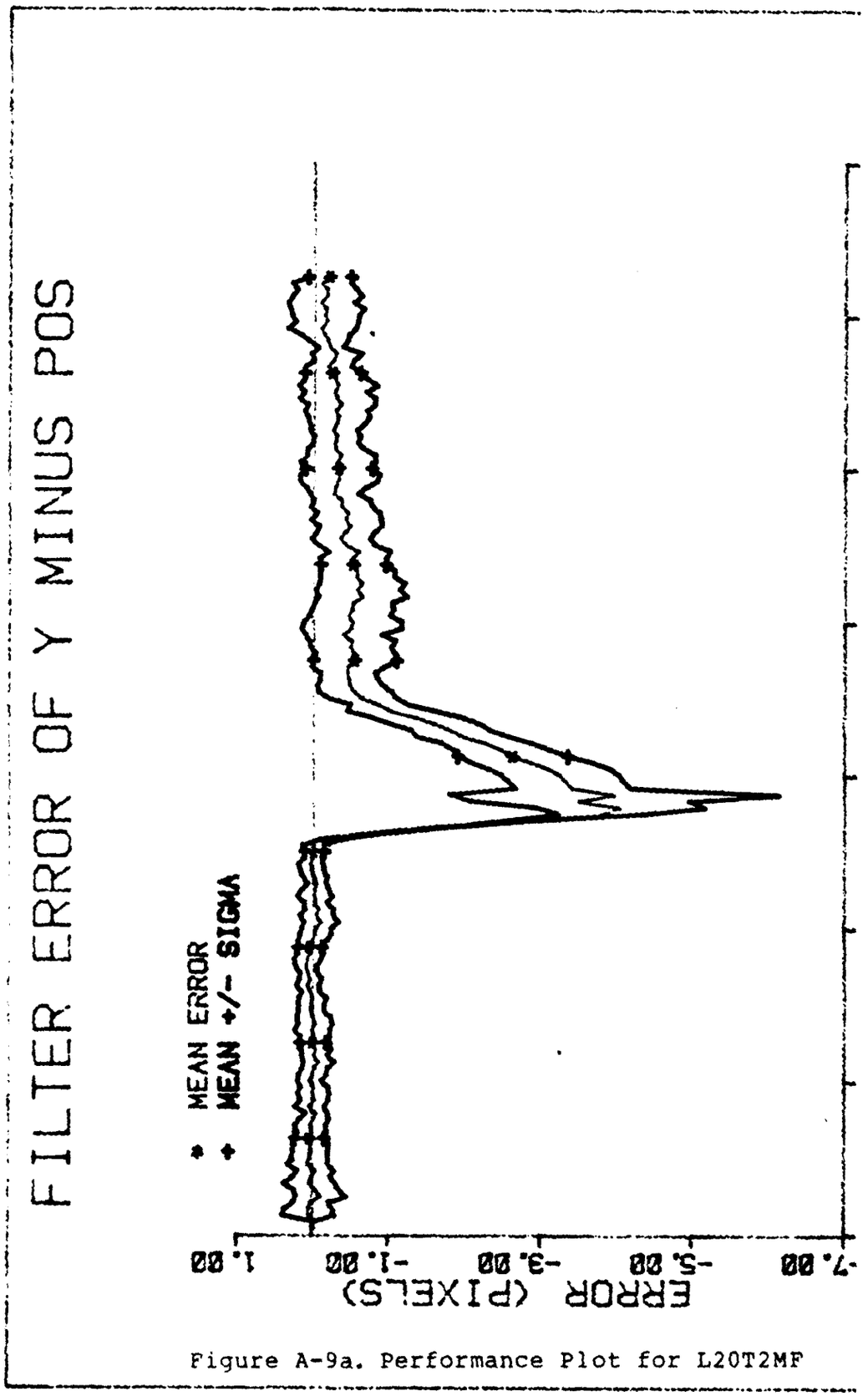


Figure A-9a. Performance Plot for L20T2MF

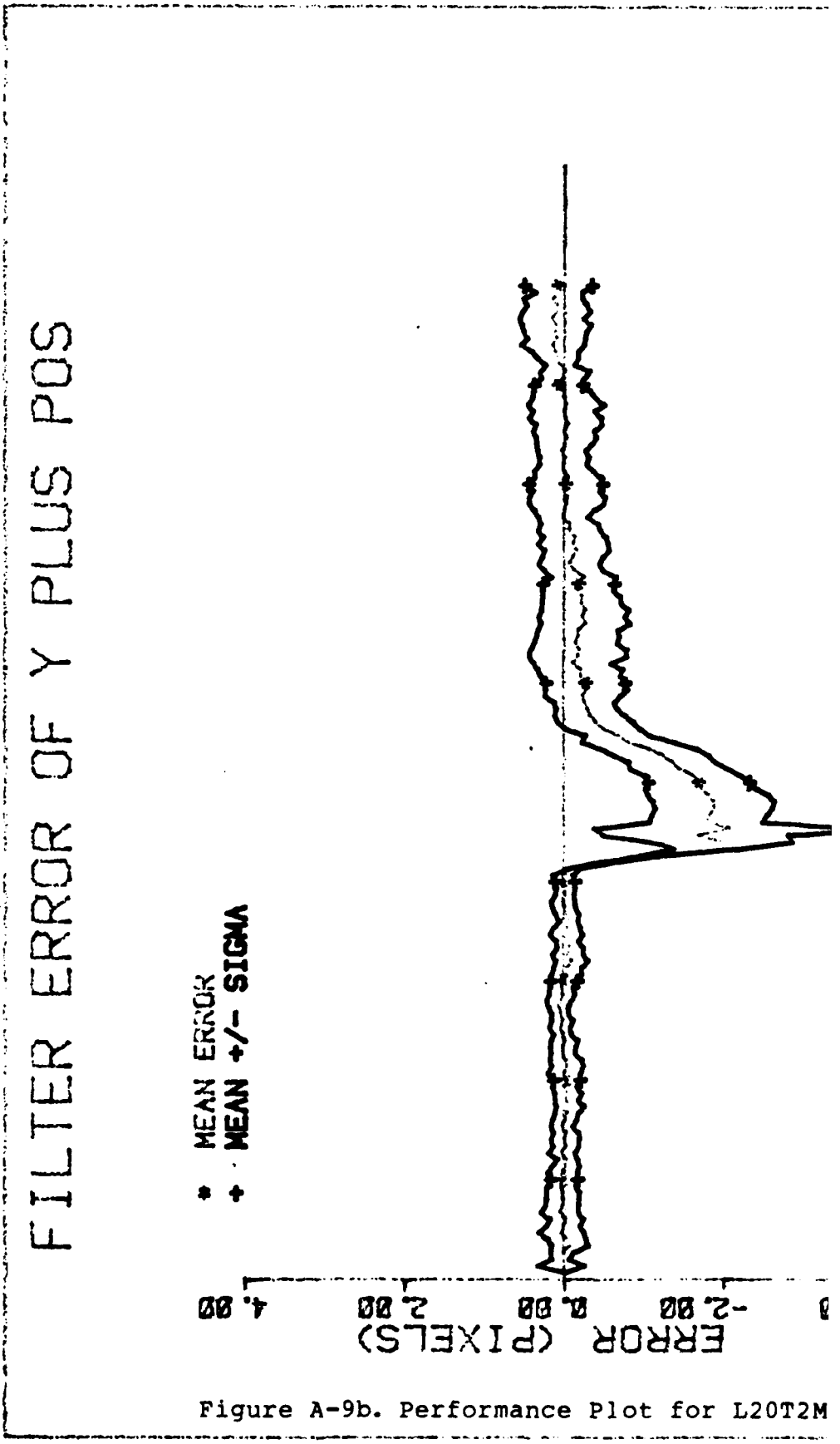


Figure A-9b. Performance Plot for L20T2M

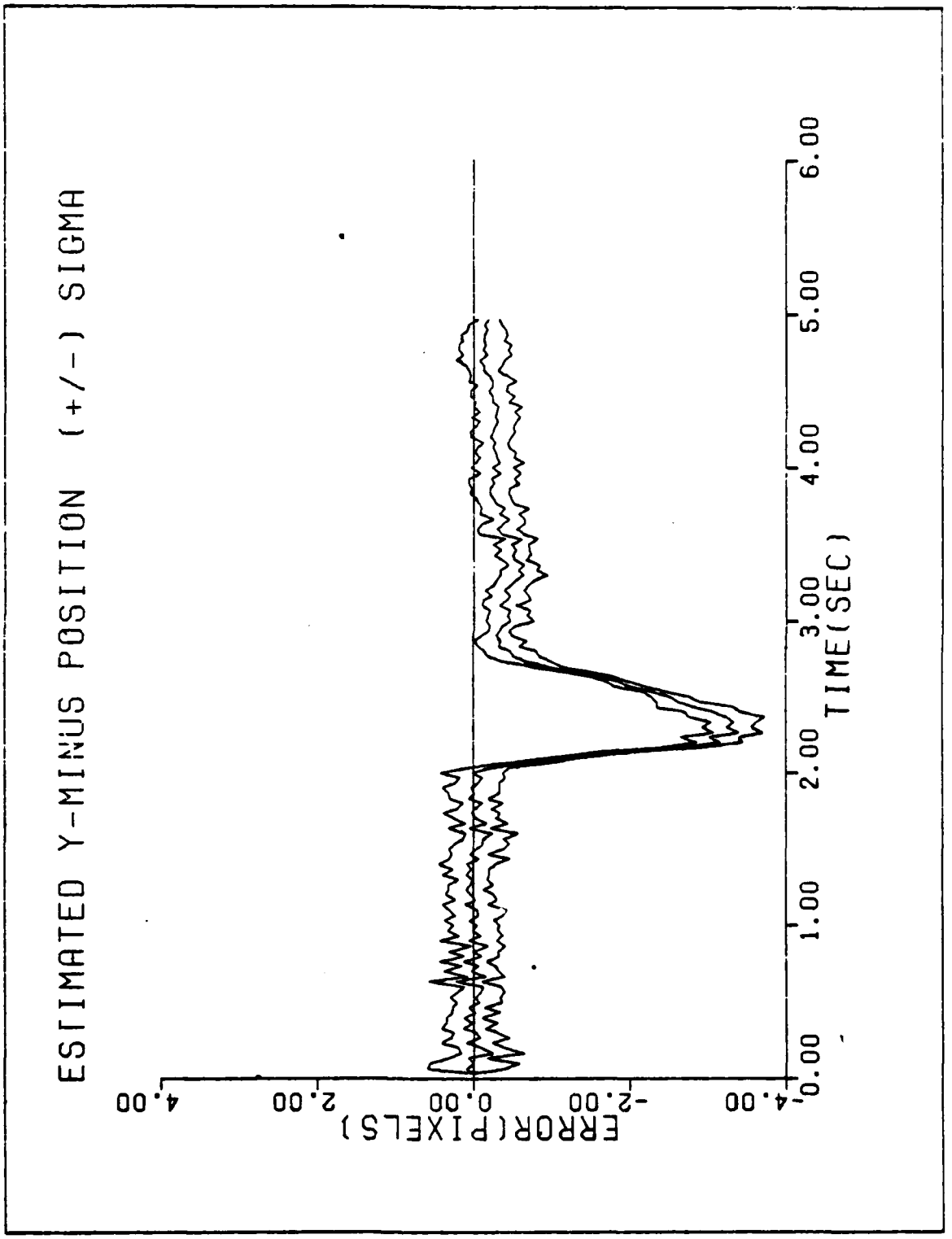


Figure A-10a. Performance Plot for L20T2F2

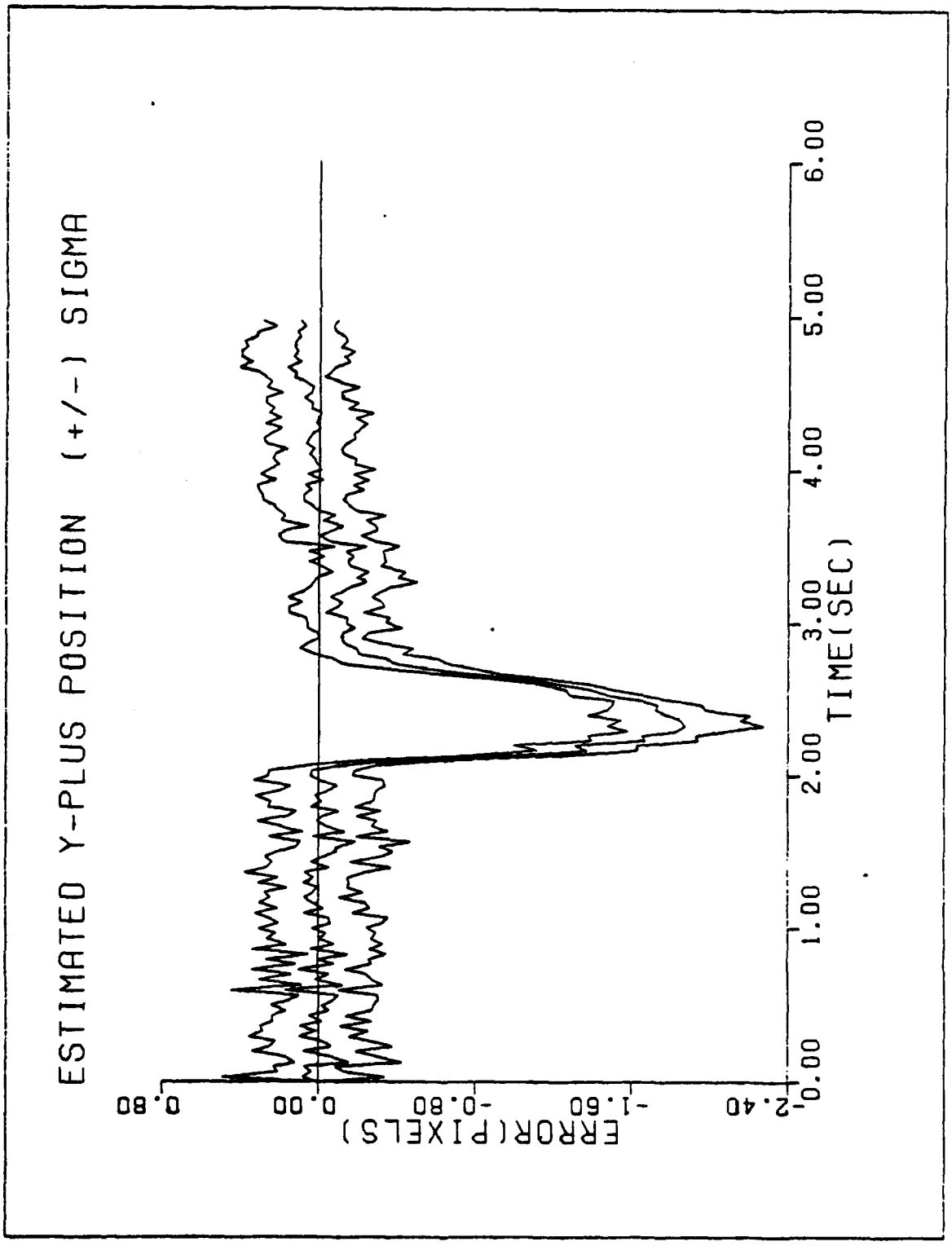


Figure A-10b. Performance Plot for L20T2F2

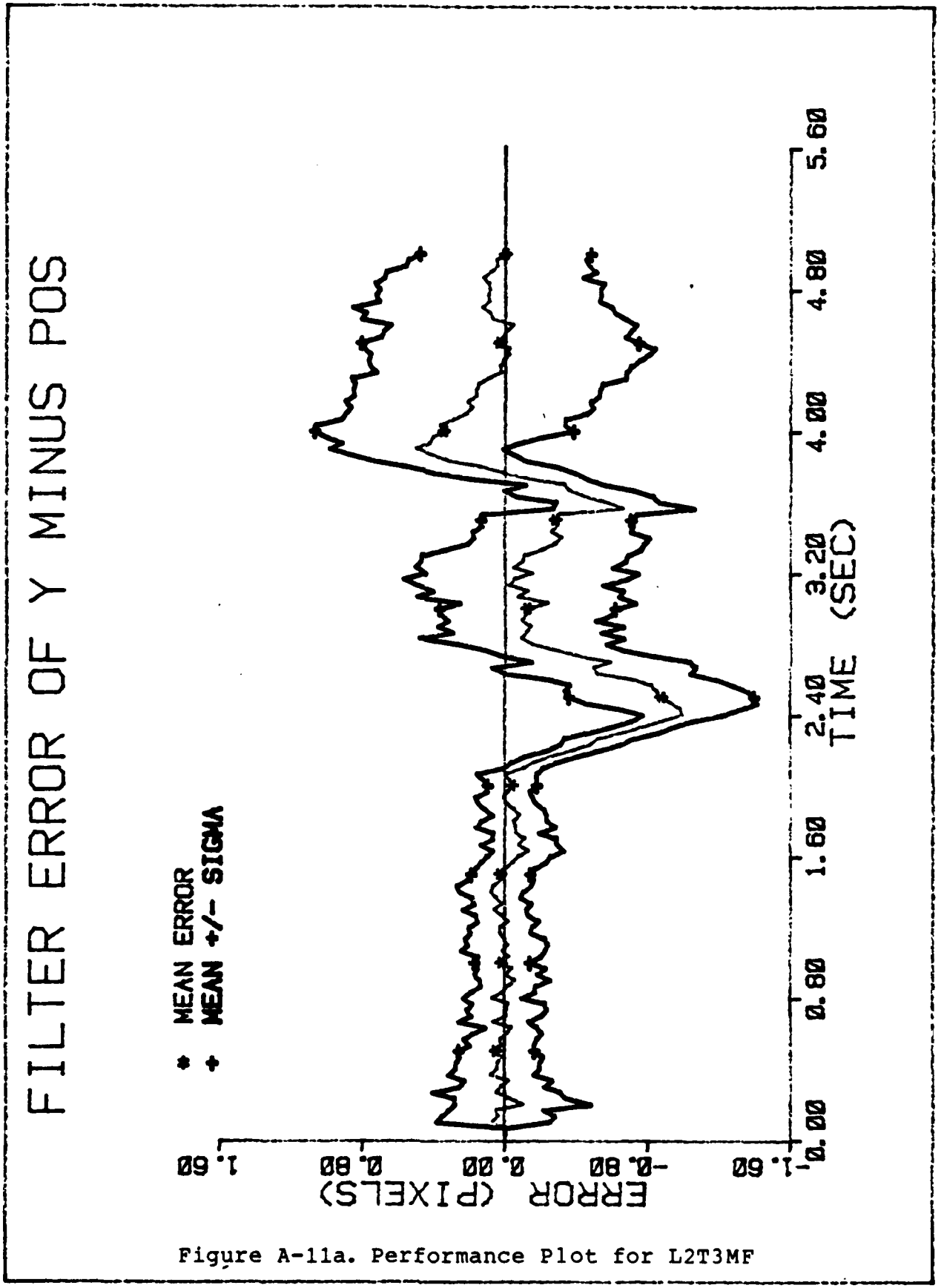


Figure A-11a. Performance Plot for L2T3MF

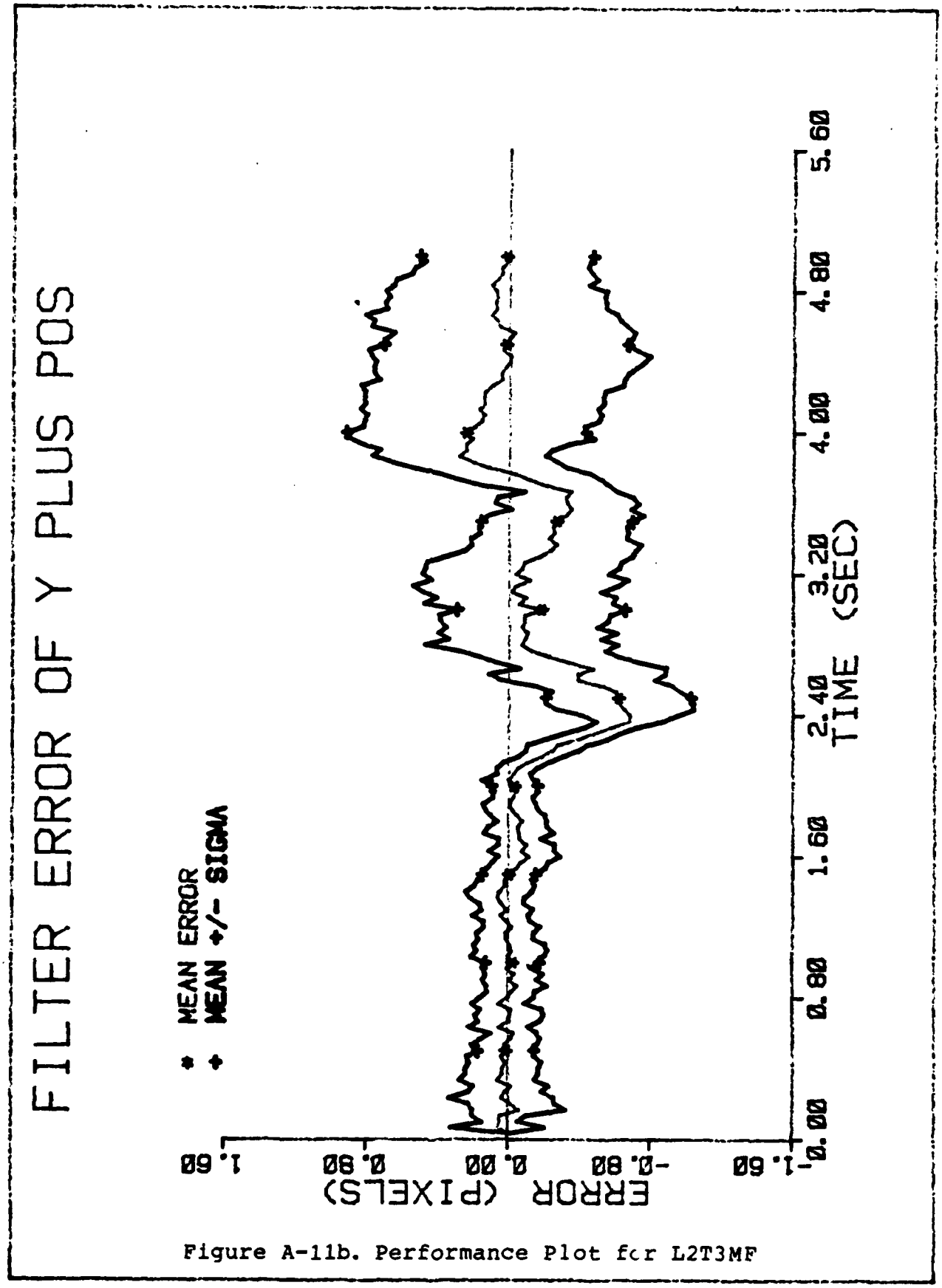


Figure A-11b. Performance Plot for L2T3MF

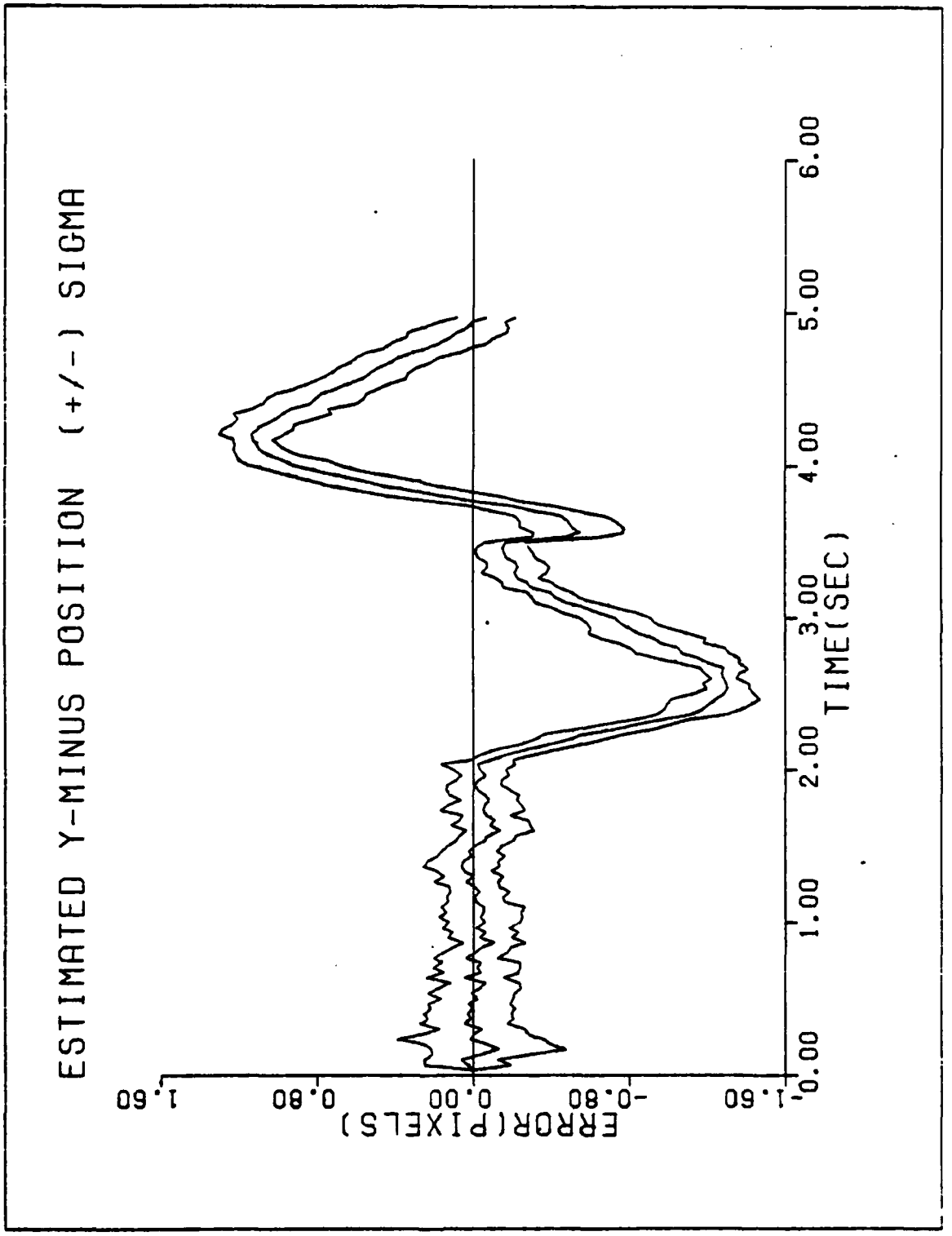


Figure A-12a. Performance Plot for L2T3F1

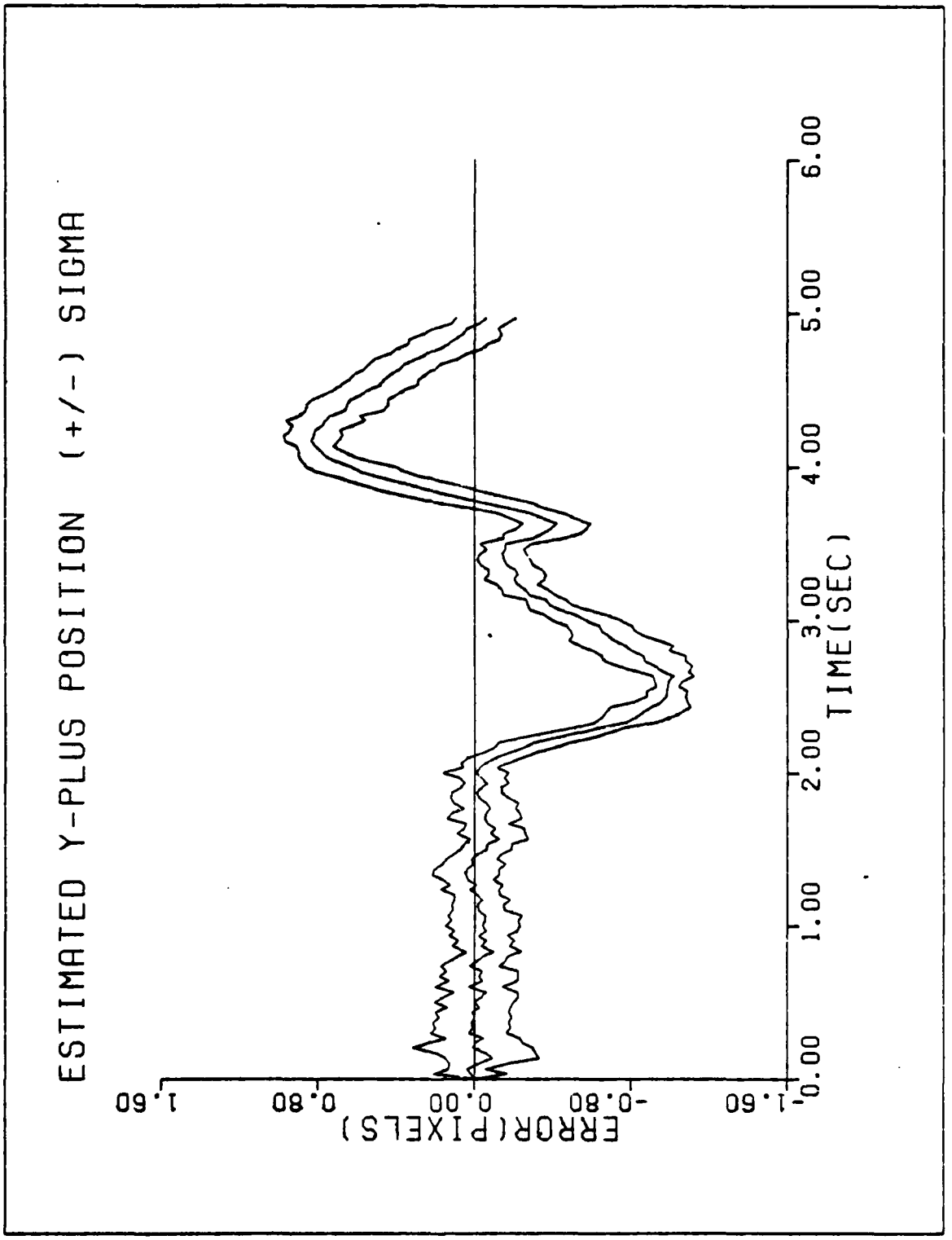


Figure A-12b. Performance Plot for L2T3F1

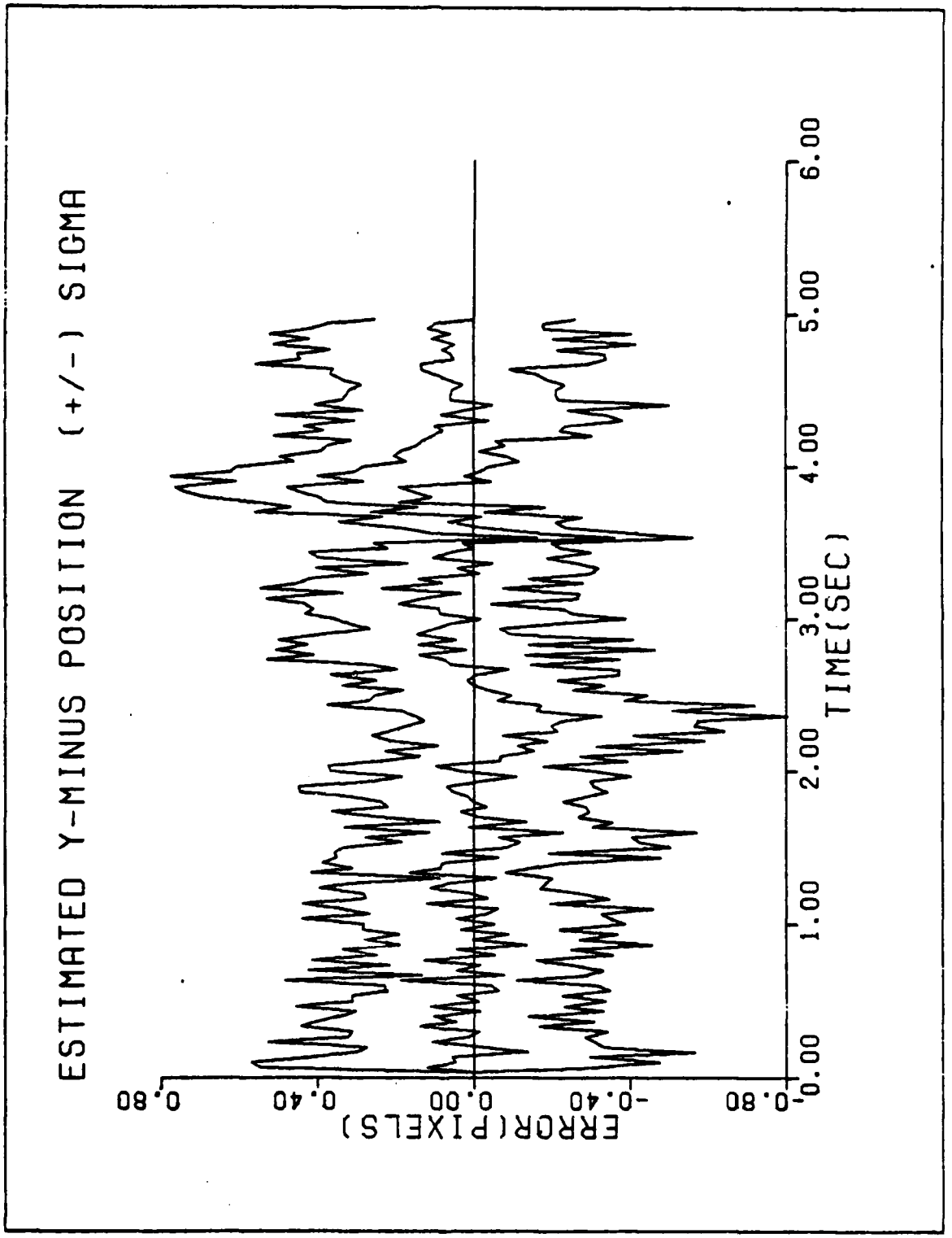


Figure A-13a. Performance Plot for L2T3F2

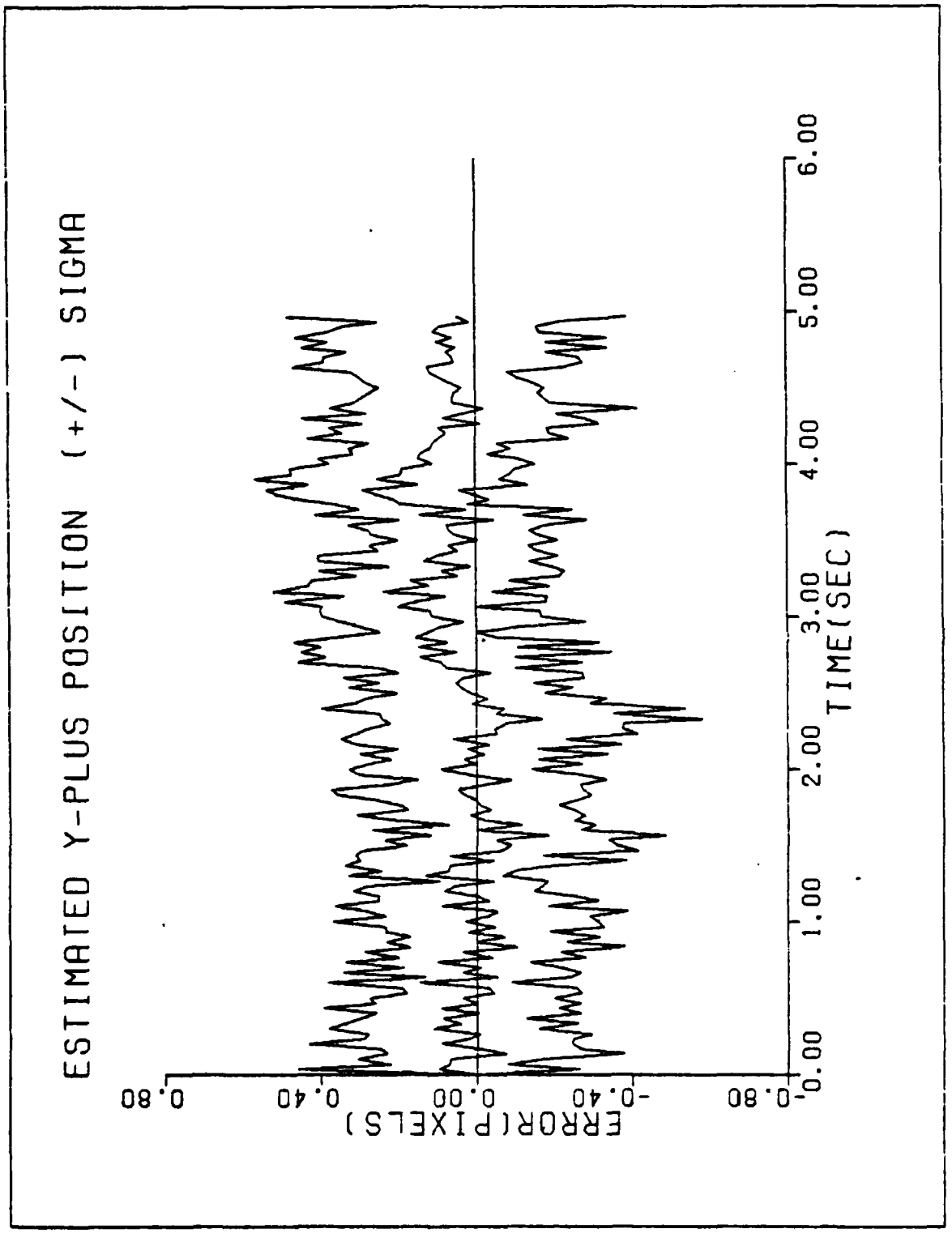


Figure A-13b. Performance Plot for L2T3F2

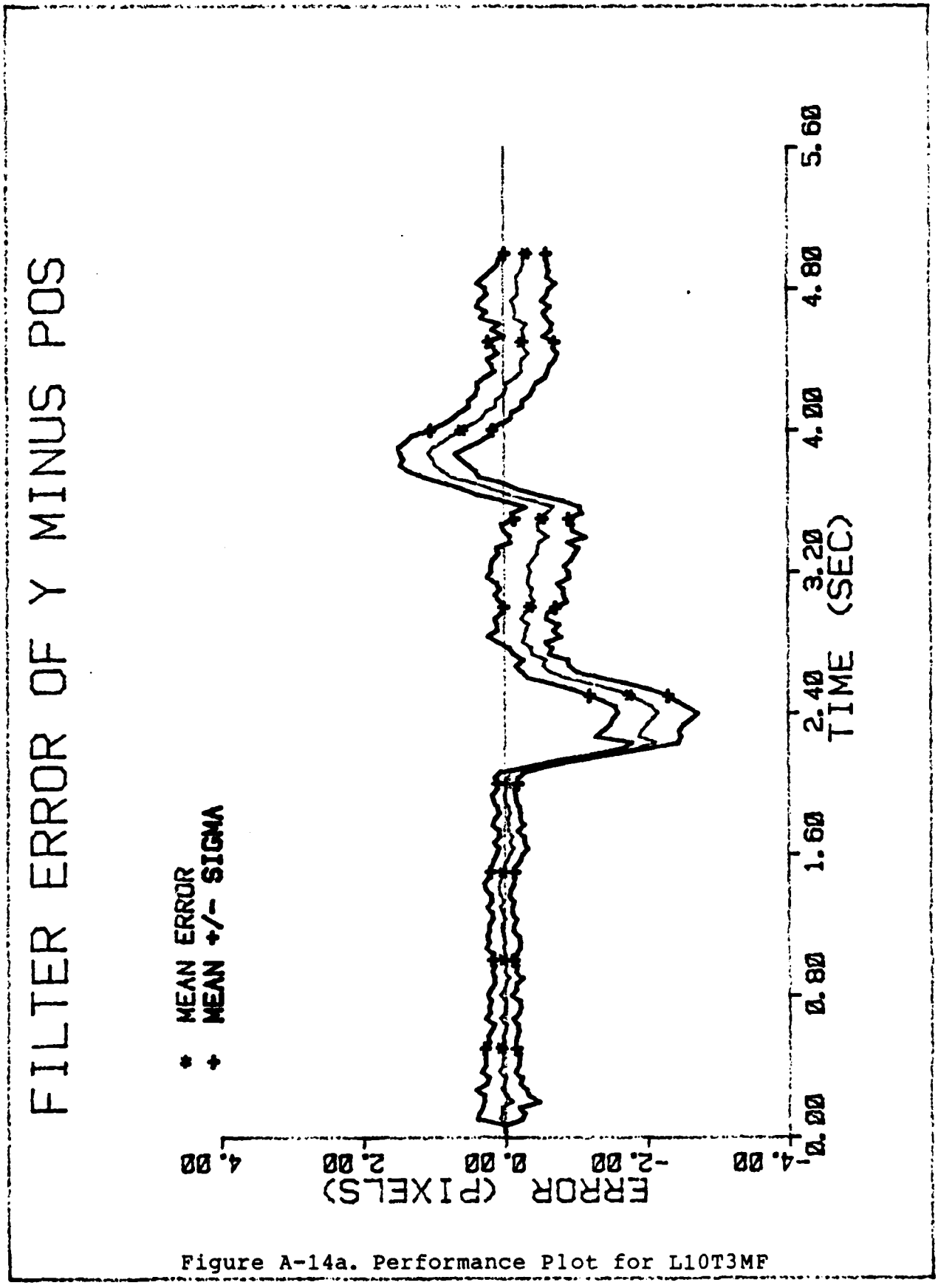


Figure A-14a. Performance Plot for L10T3MF

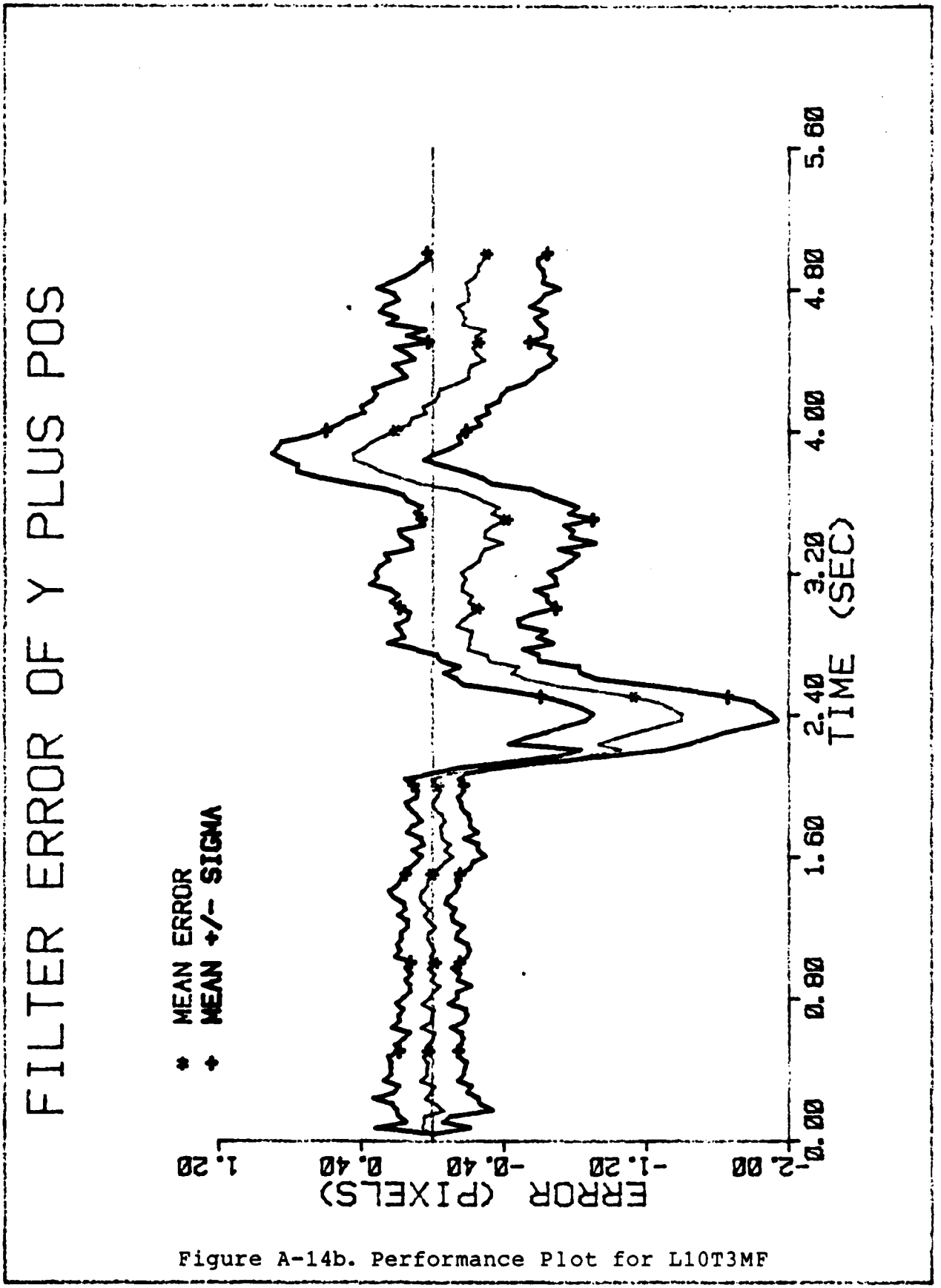


Figure A-14b. Performance Plot for L10T3MF

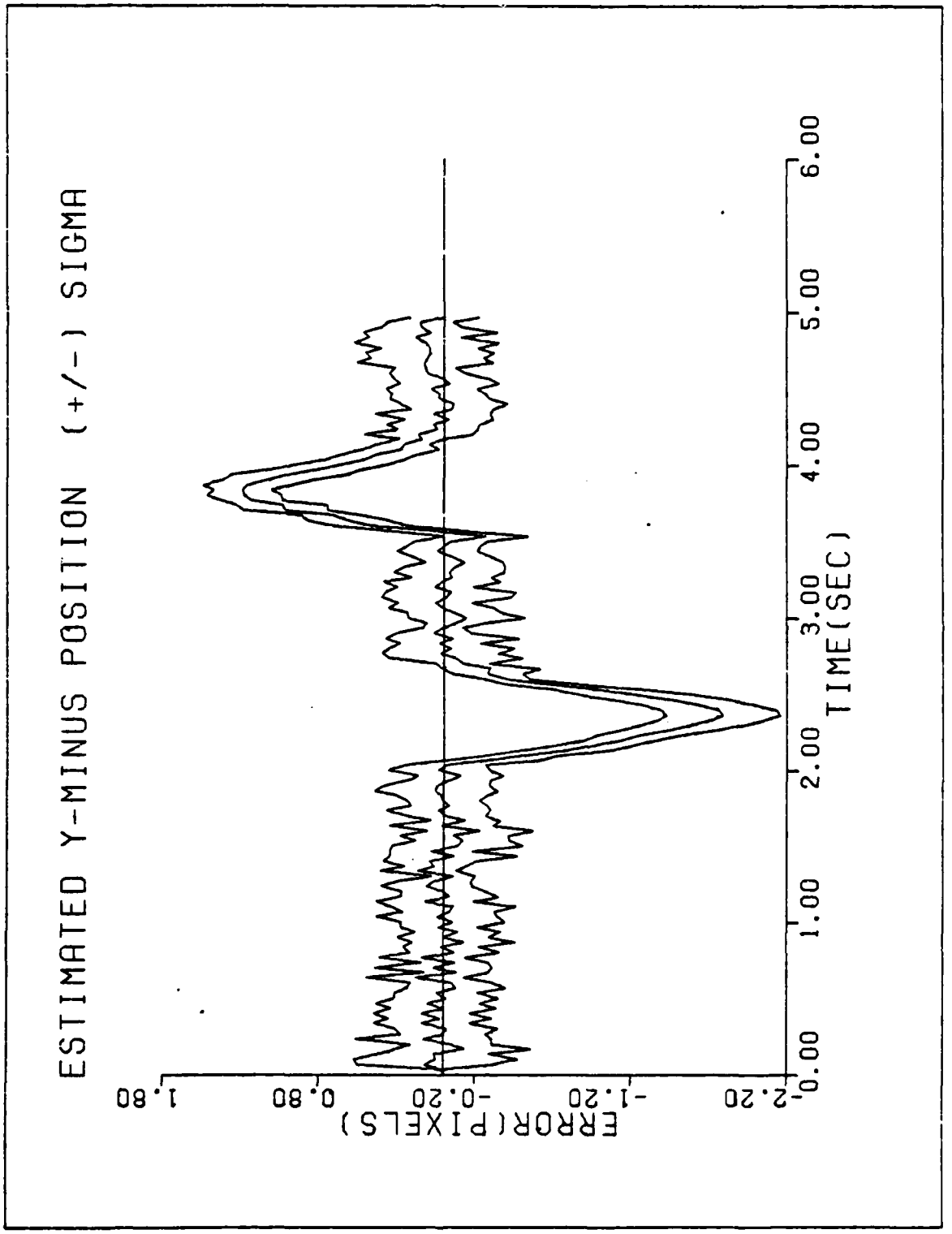


Figure A-15a. Performance Plot for L10T3F2

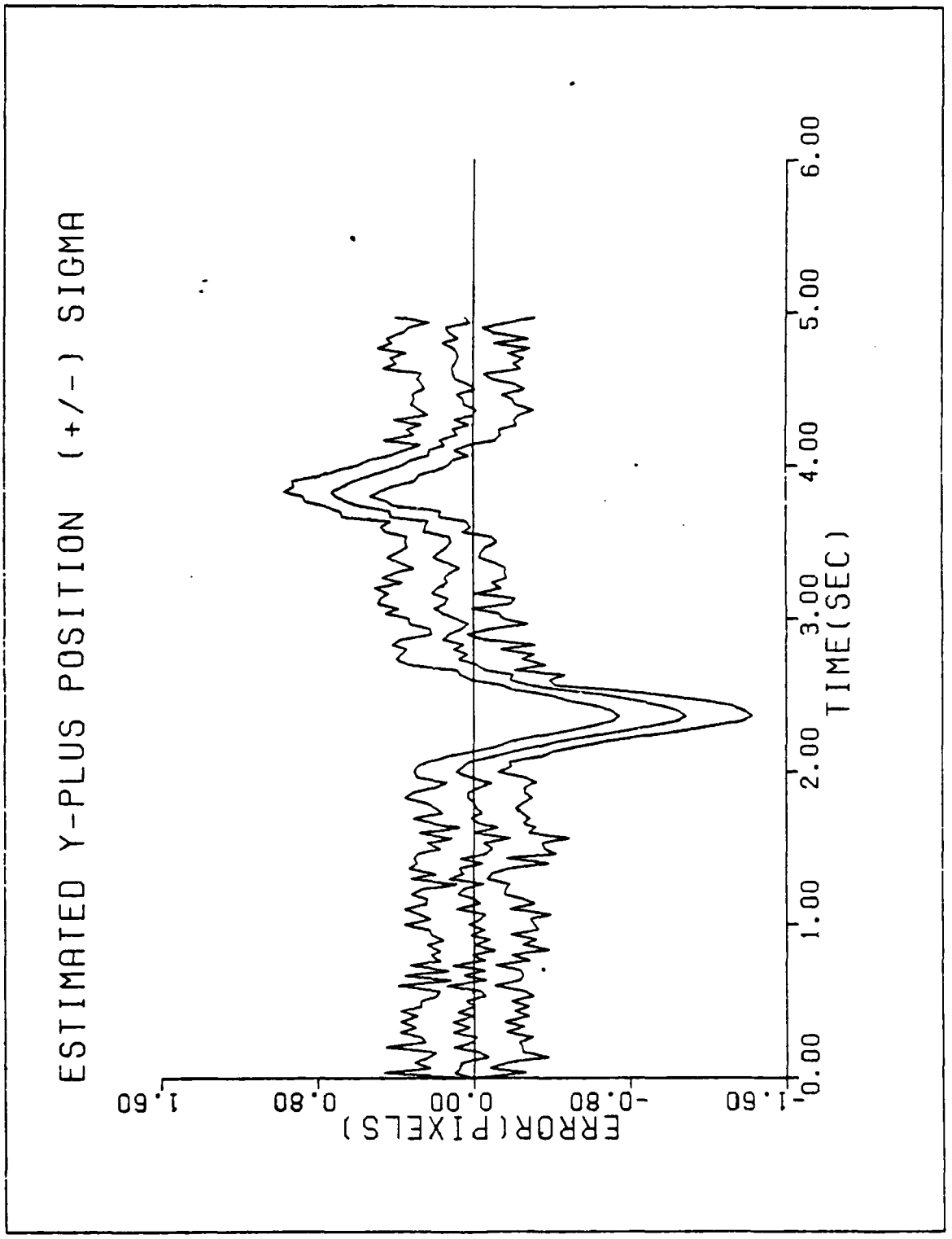


Figure A-15b. Performance Plot for L10T3F2

Figure A-16a. Performance

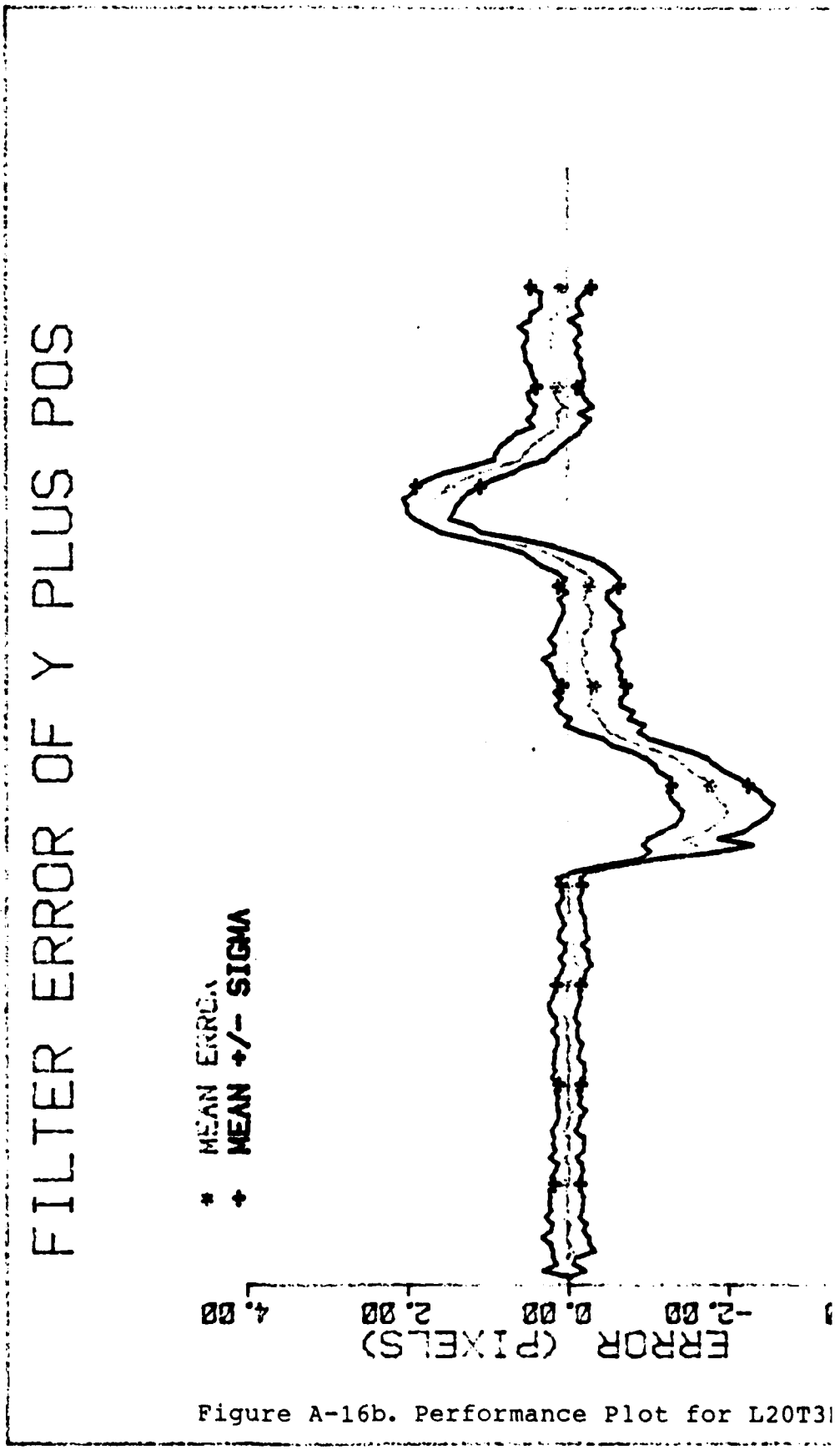


Figure A-16b. Performance Plot for L20T3

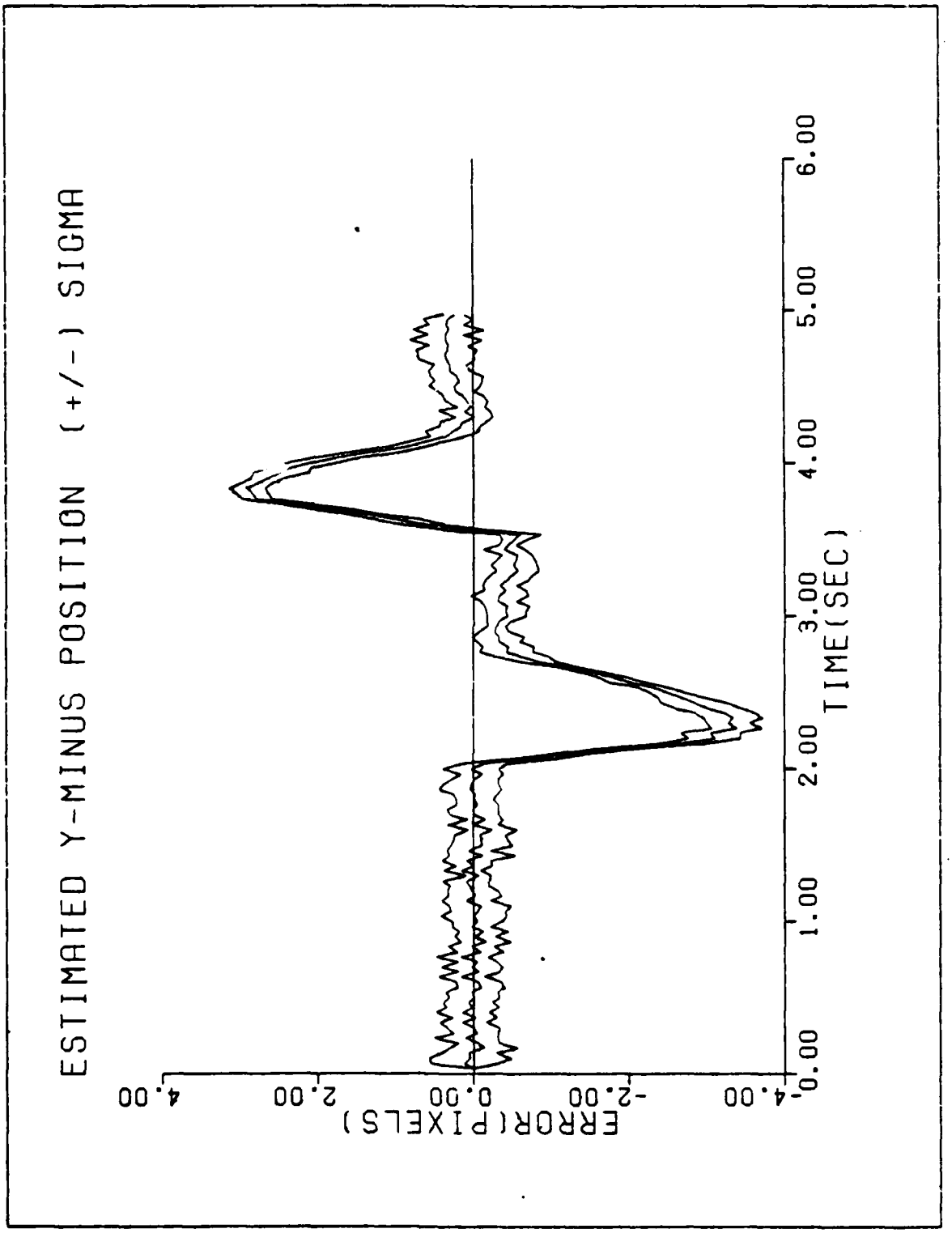


Figure A-17a. Performance Plot for L20T3F2

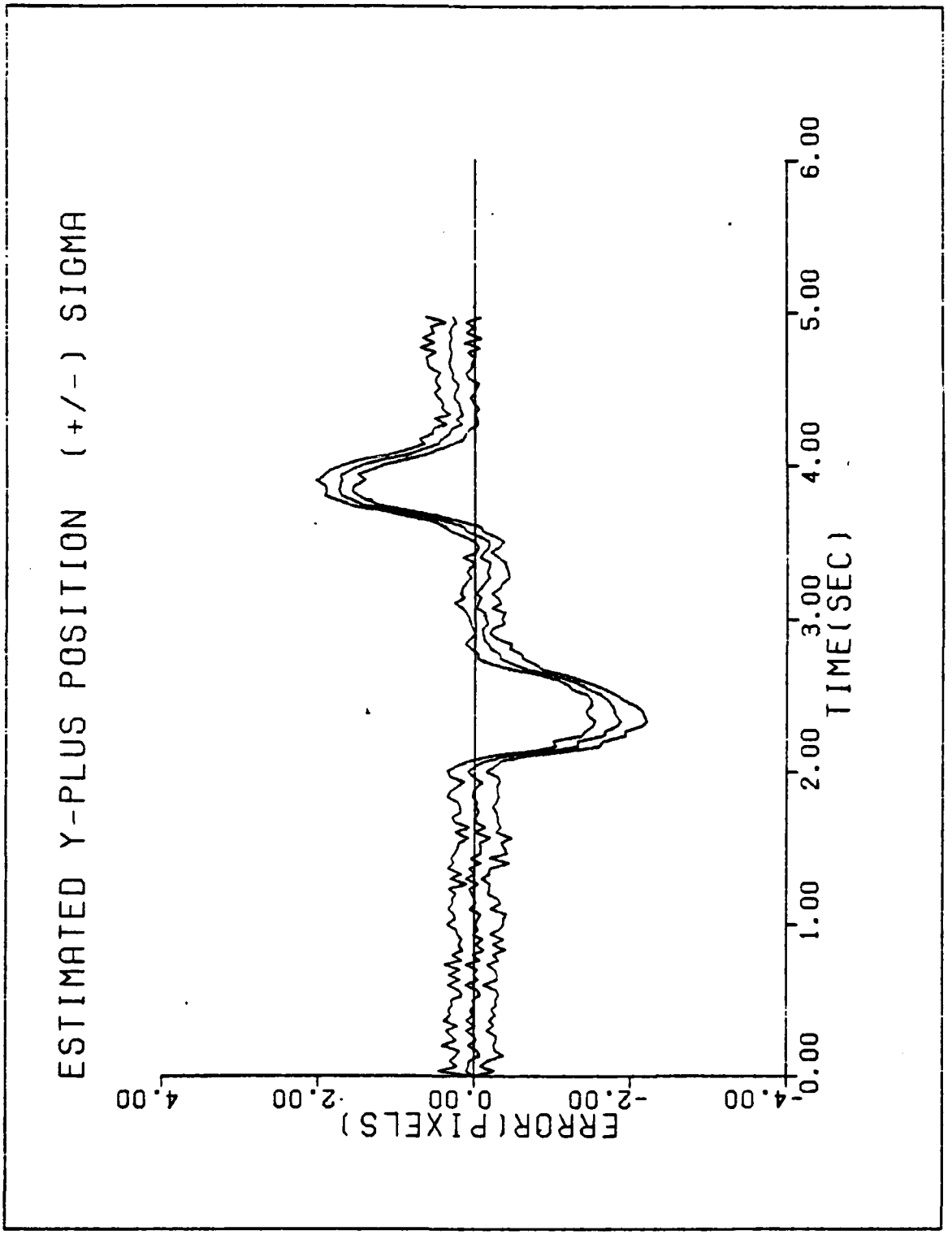


Figure A-17b. Performance Plot for L20T3F2

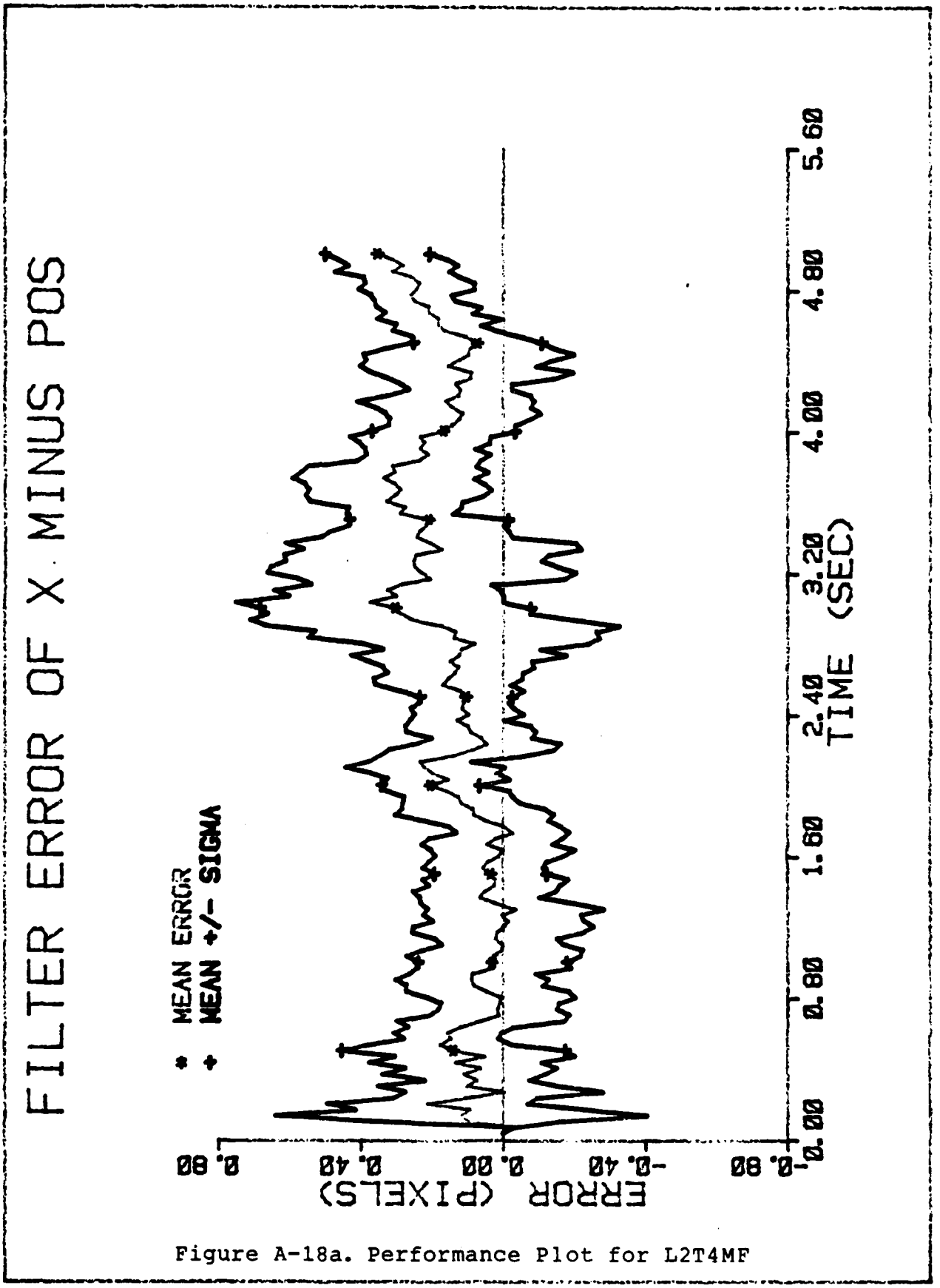


Figure A-18a. Performance Plot for L2T4MF

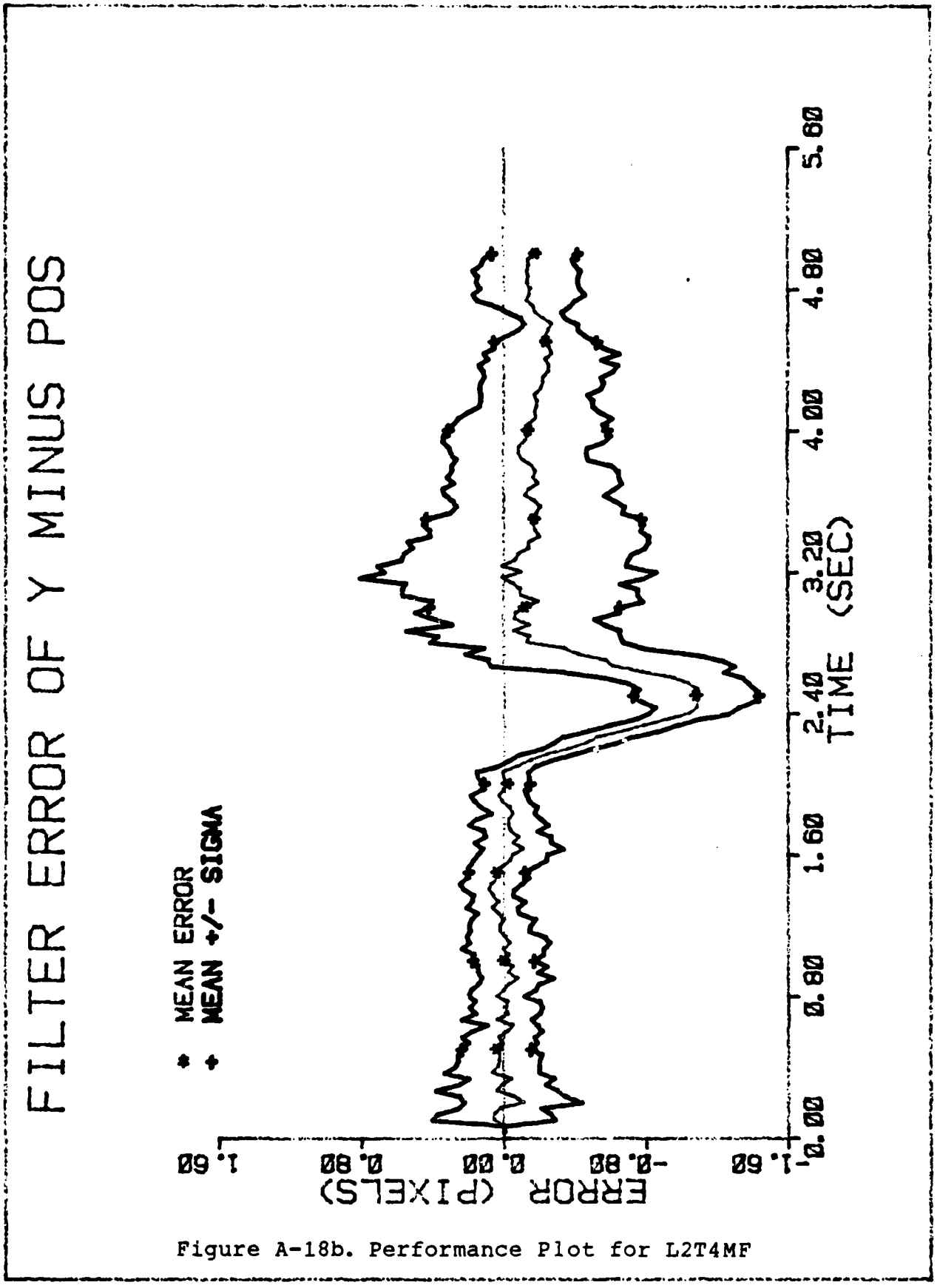


Figure A-18b. Performance Plot for L2T4MF

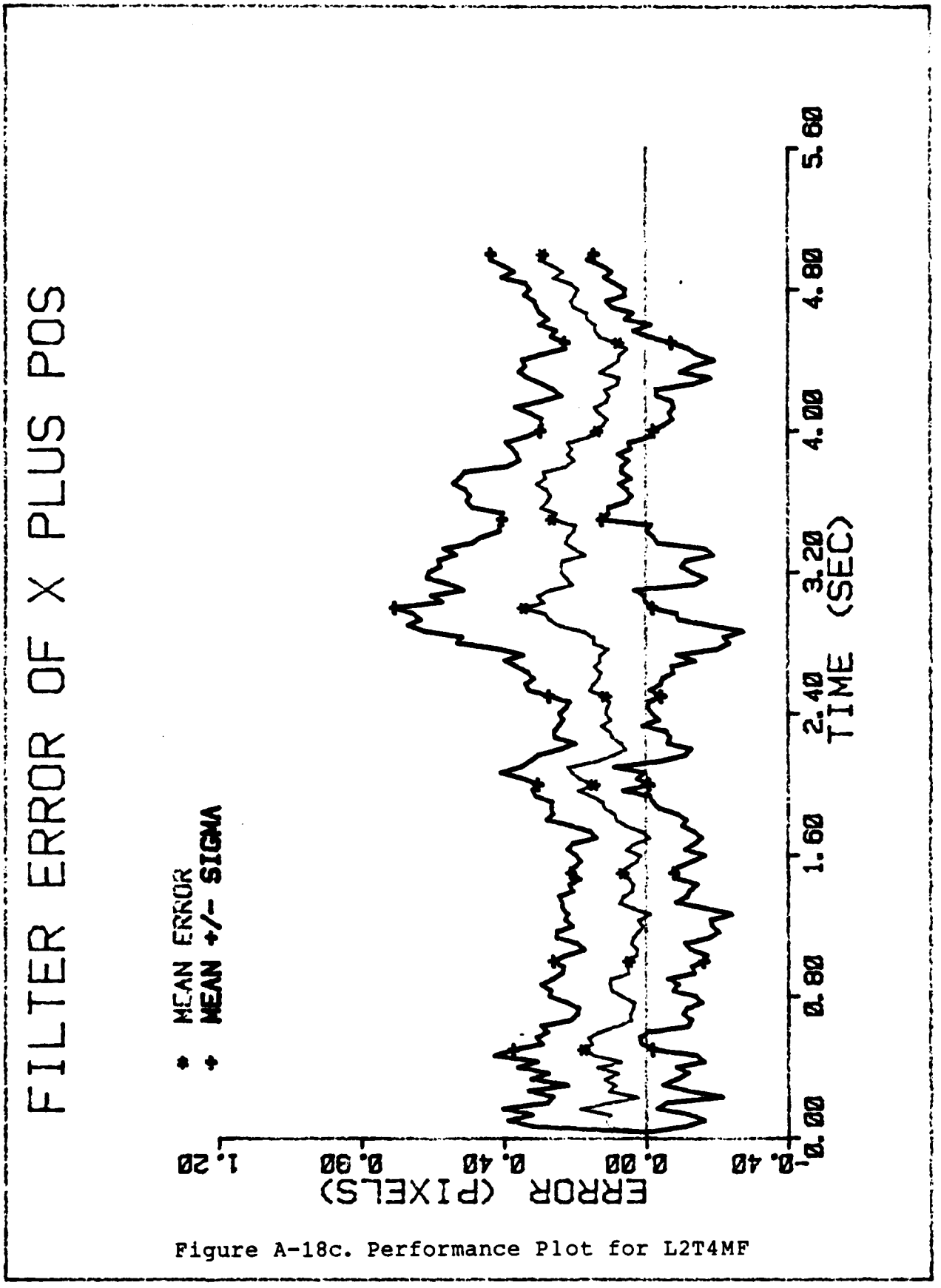


Figure A-18c. Performance Plot for L2T4MF

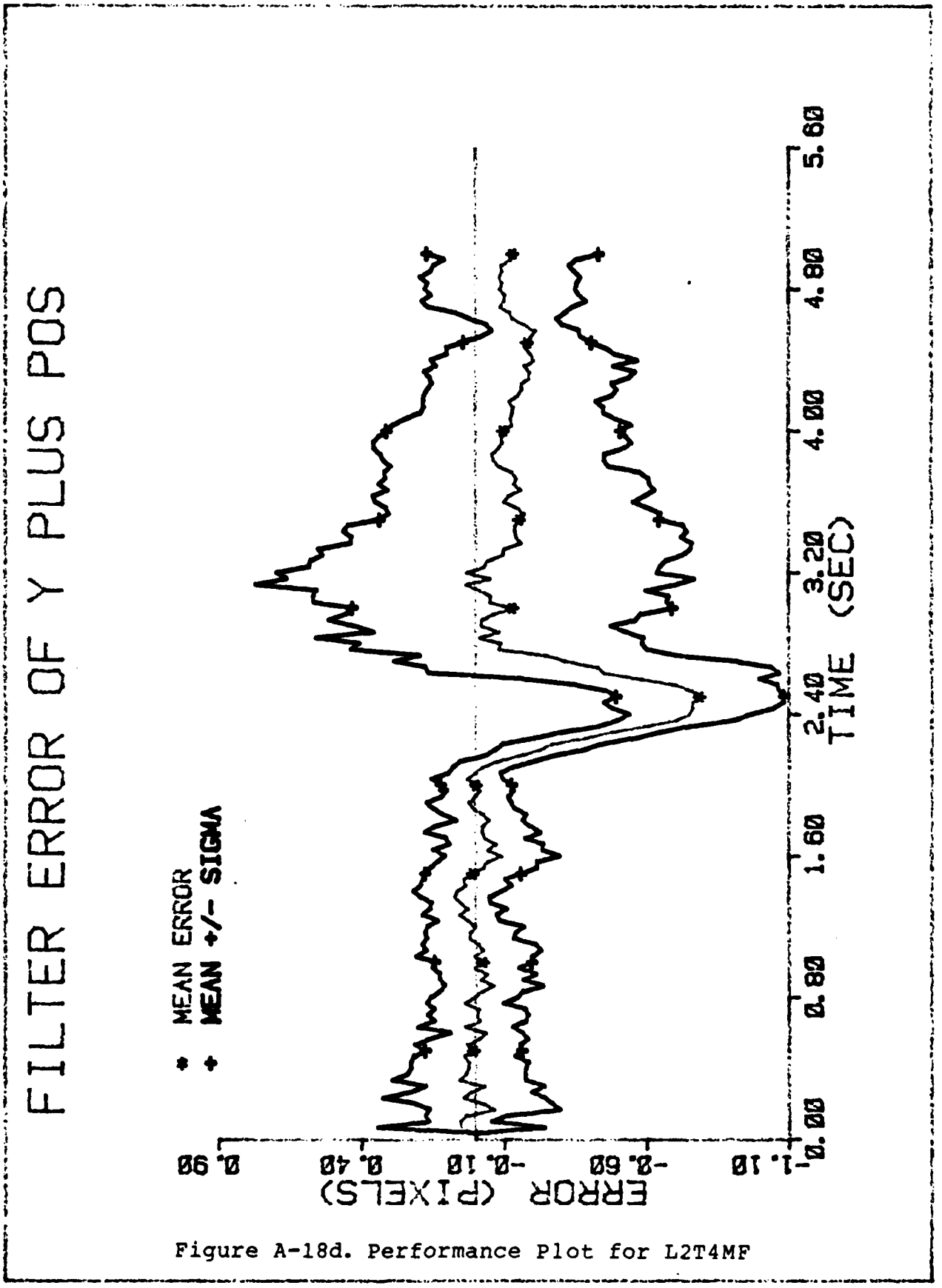


Figure A-18d. Performance Plot for L2T4MF

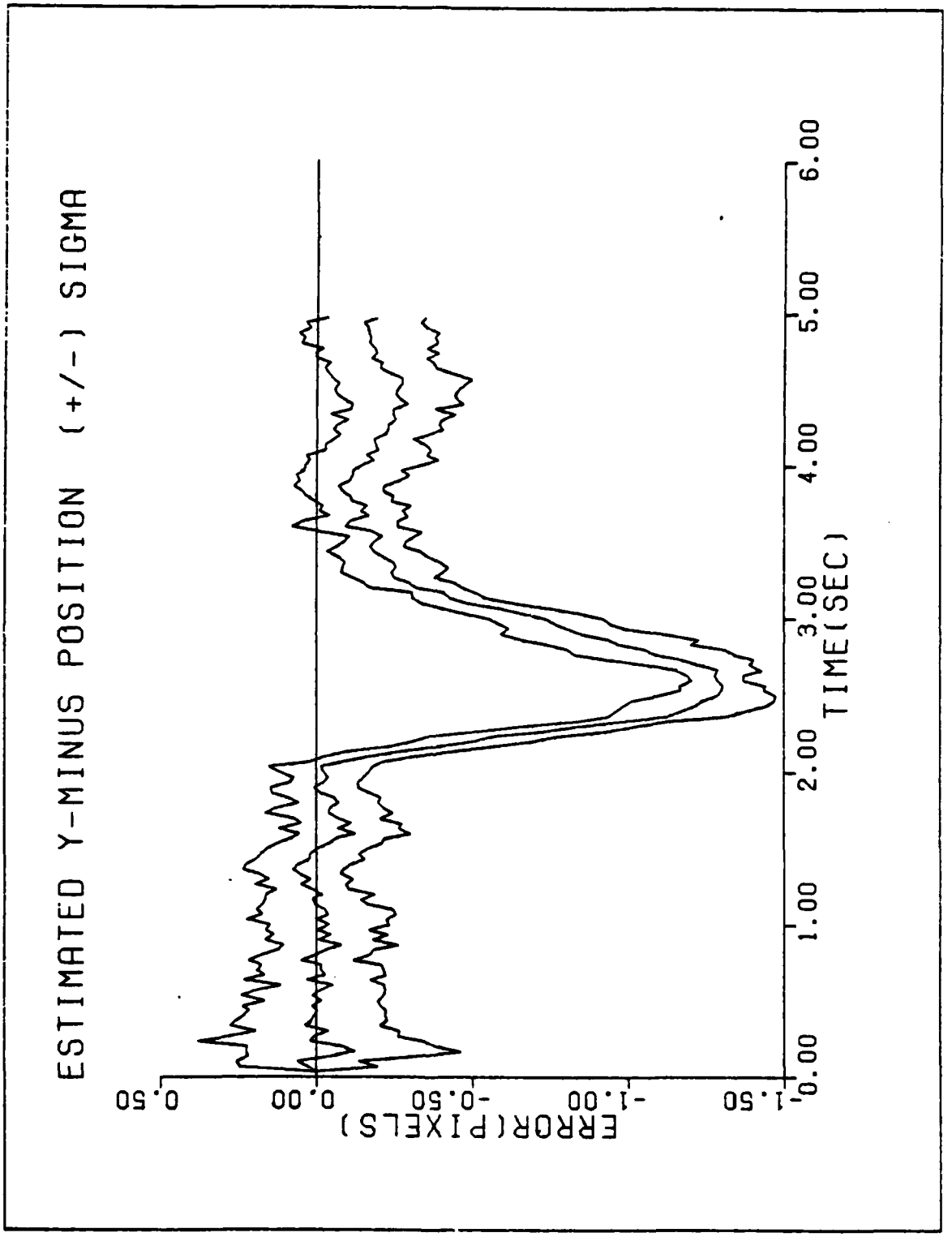


Figure A-19a. Performance Plot for L2T4F1

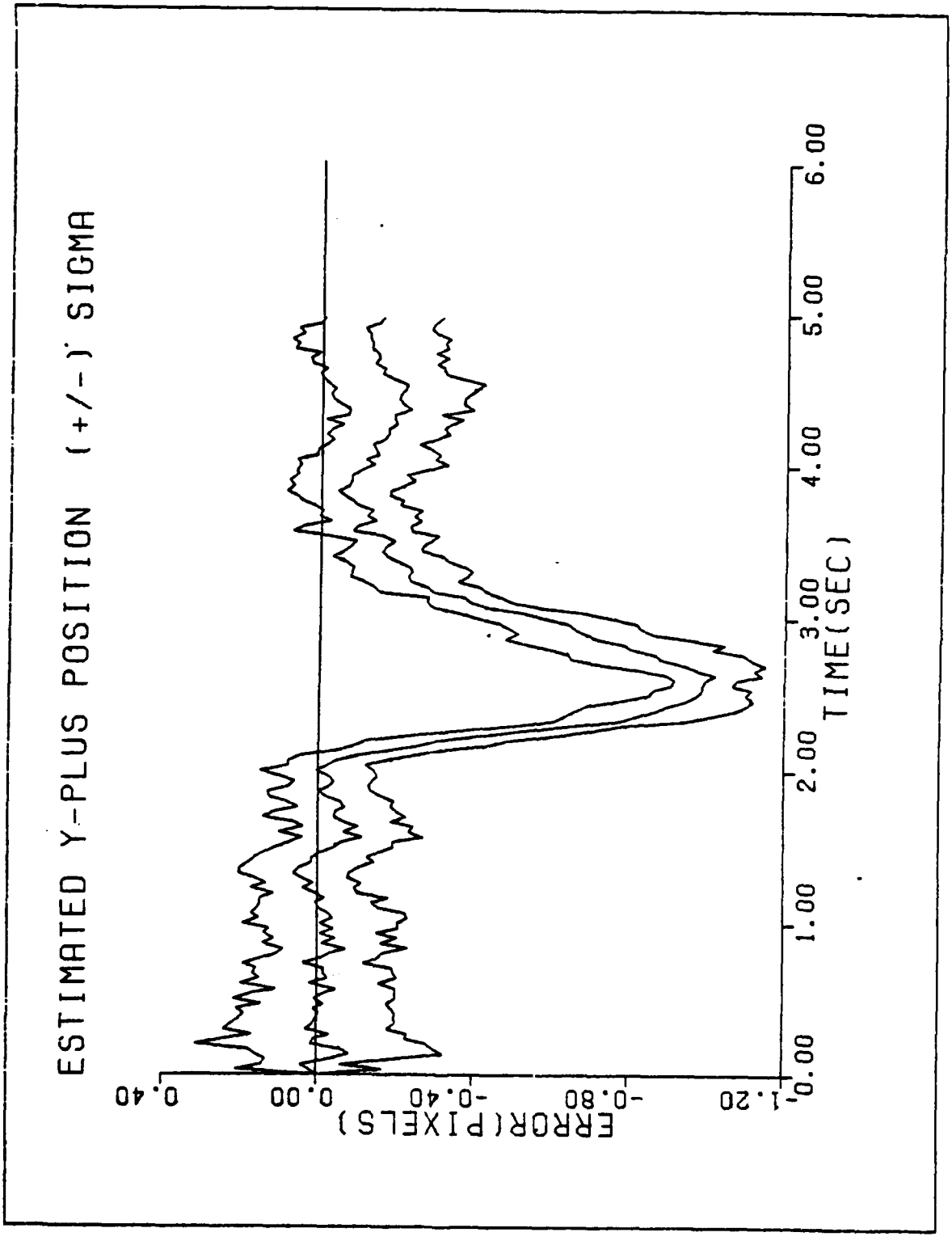


Figure A-19b. Performance Plot for L2T4F1

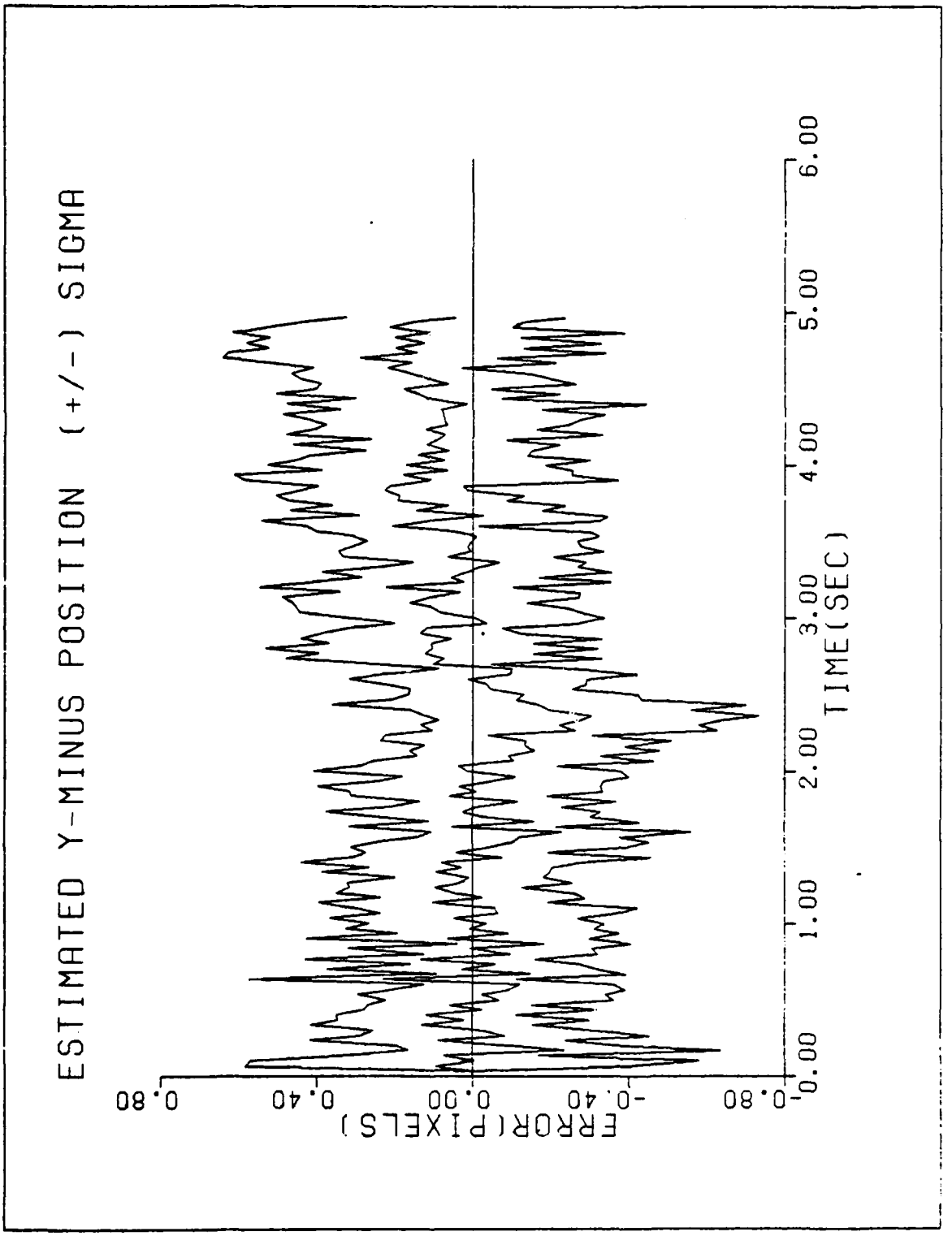


Figure A-20a. Performance Plot for L2T4F2

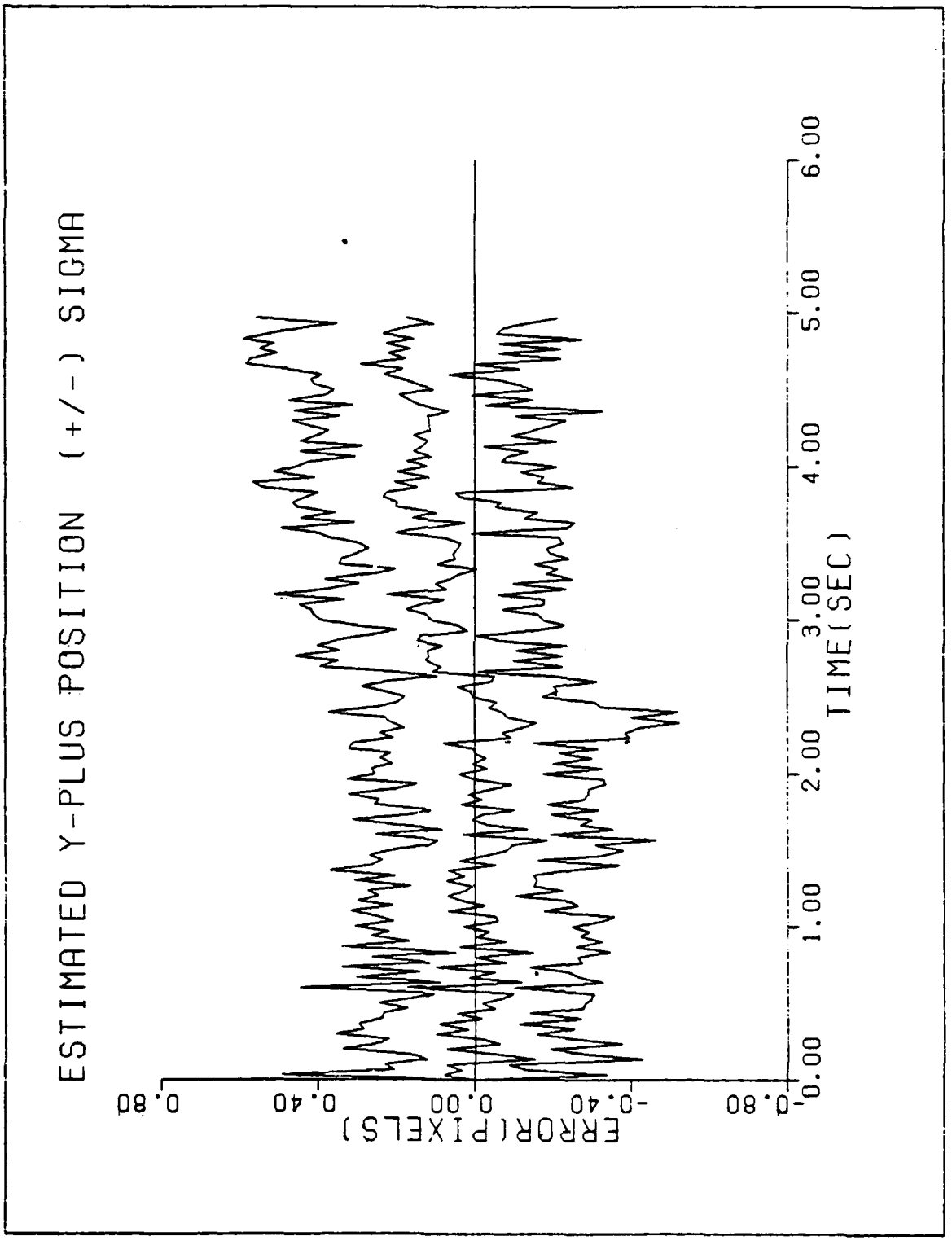


Figure A-20b. Performance Plot for L2T4F2

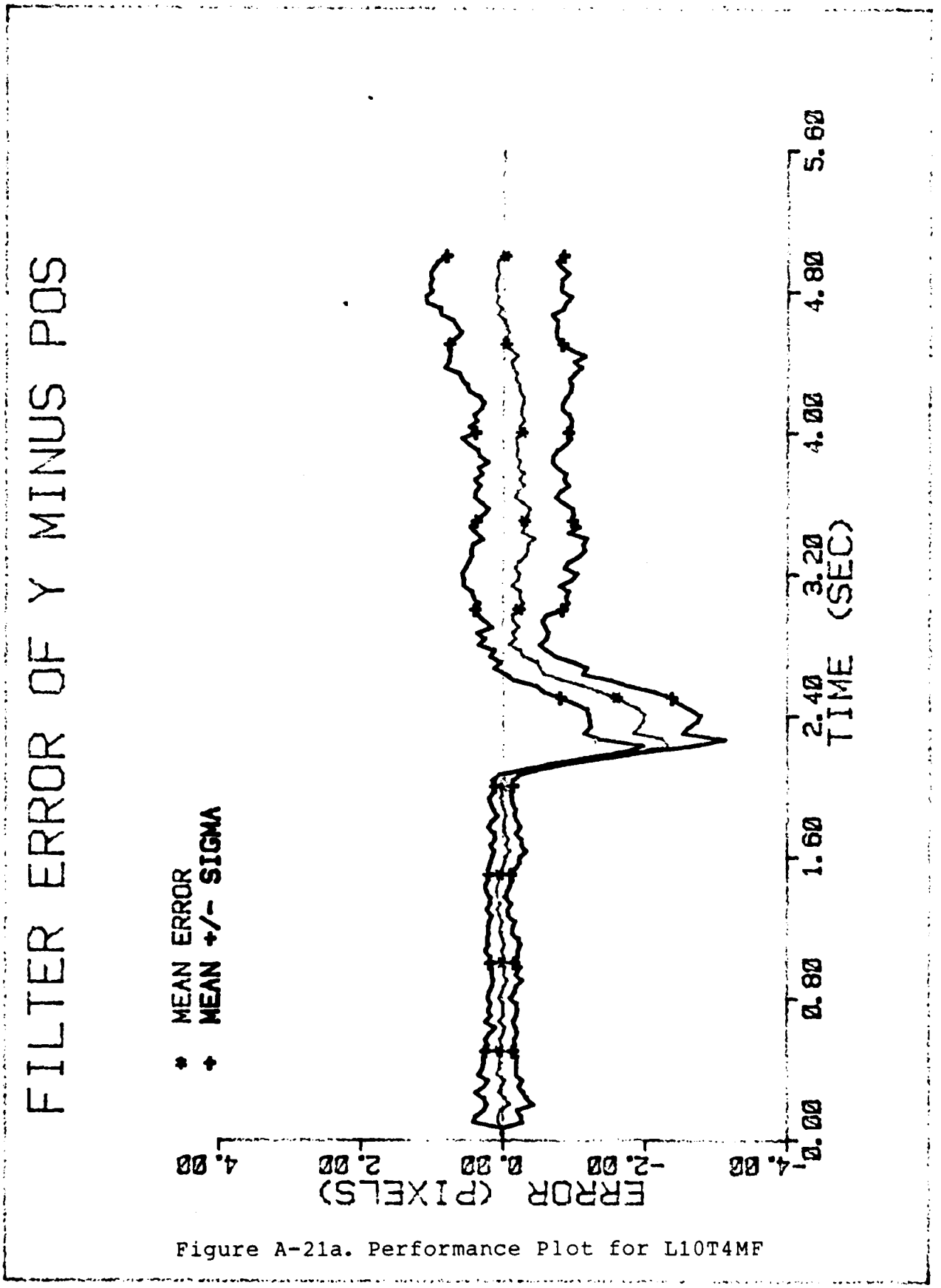


Figure A-21a. Performance Plot for L10T4MF

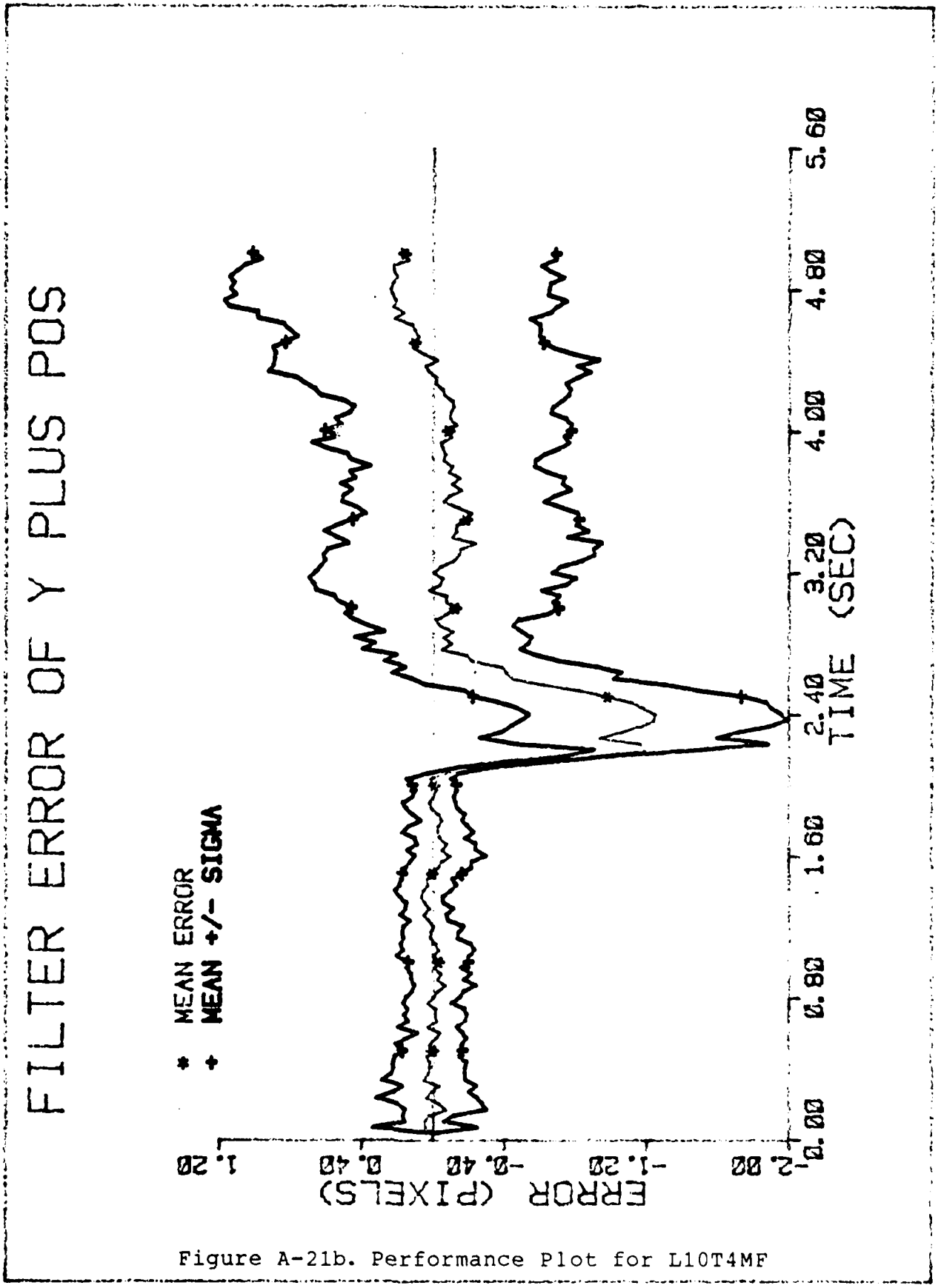
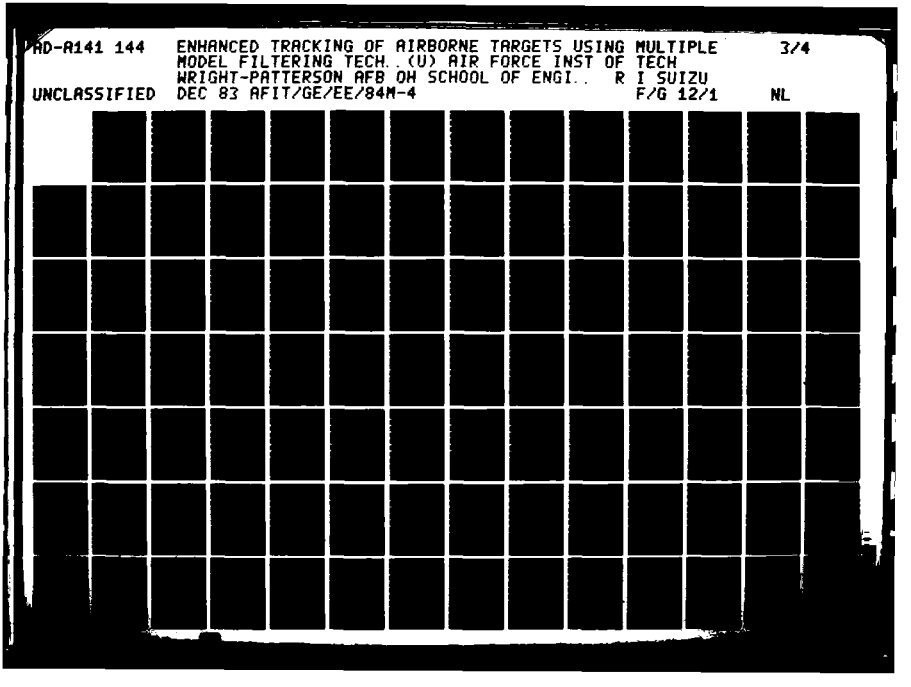
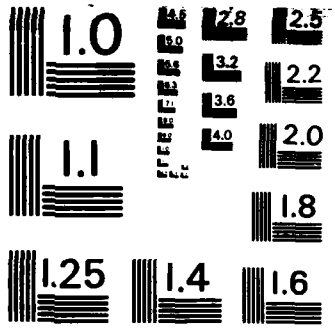


Figure A-21b. Performance Plot for L10T4MF

AD-A141 144 ENHANCED TRACKING OF AIRBORNE TARGETS USING MULTIPLE MODEL FILTERING TECH..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. R I SUIZU 3/4

UNCLASSIFIED DEC 83 AFIT/GE/EE/84M-4 F/G 12/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

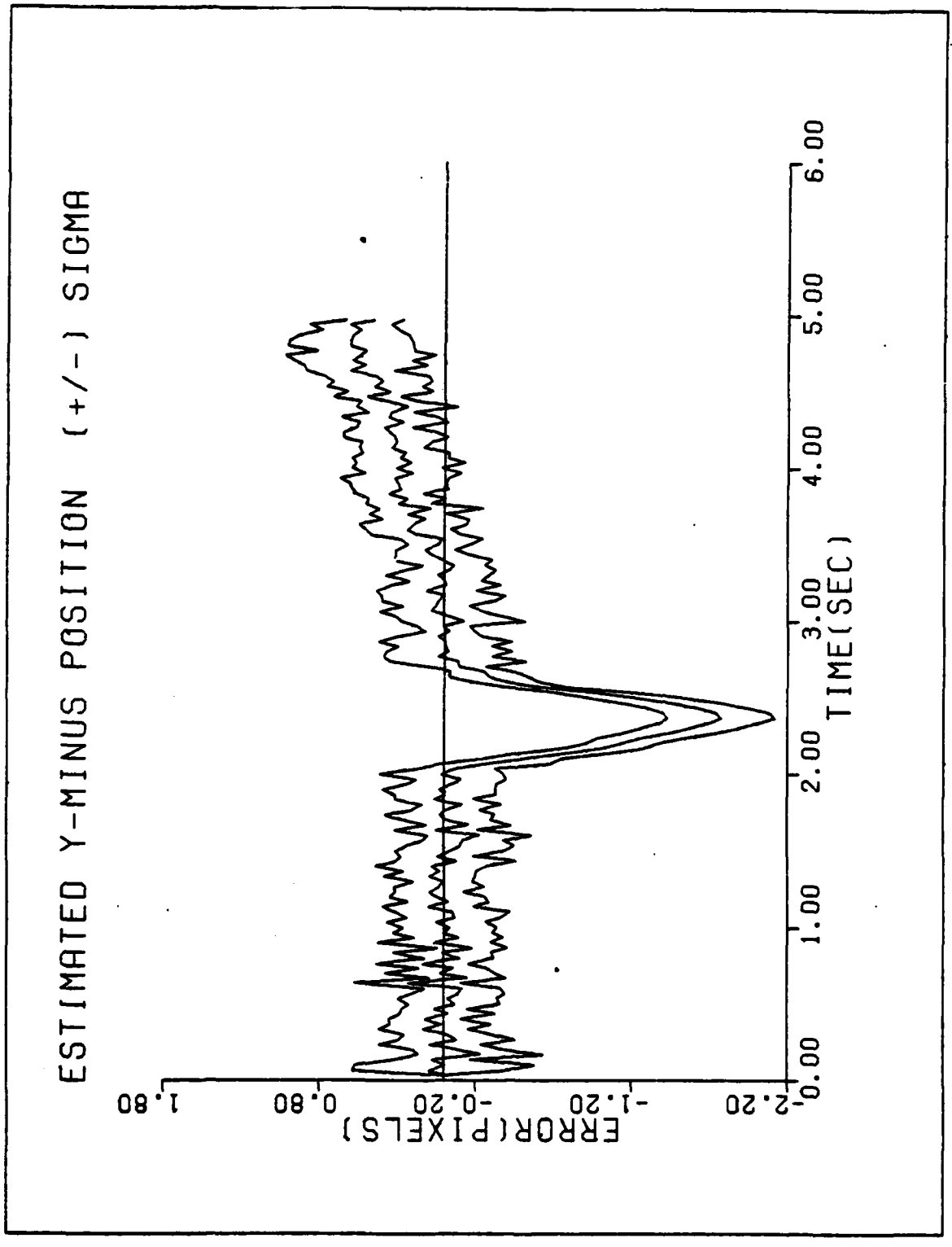


Figure A-22a. Performance Plot for L10T4F2

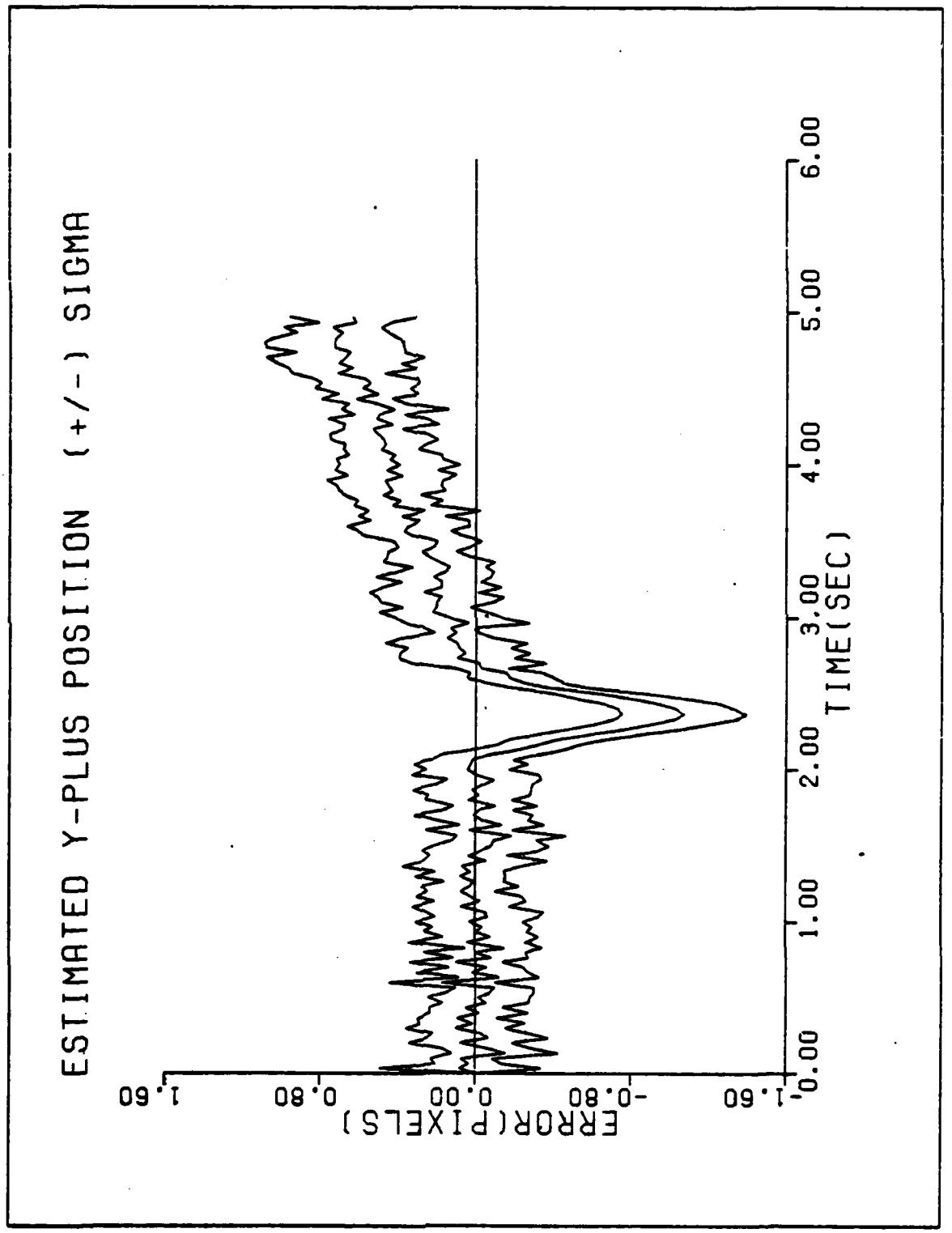


Figure A-22b. Performance Plot for L10T4F2

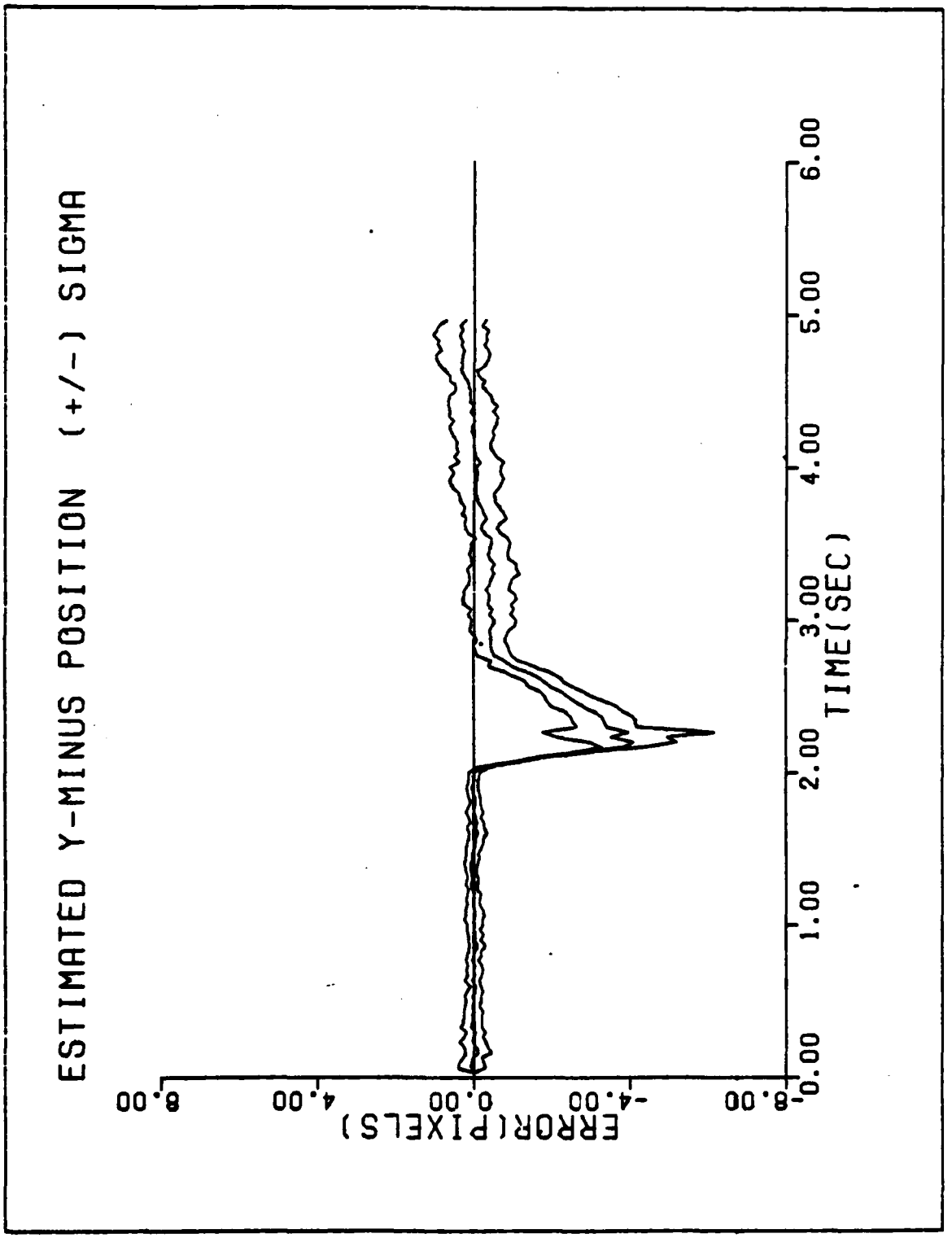


Figure A-23a. Performance Plot for L20T4MF

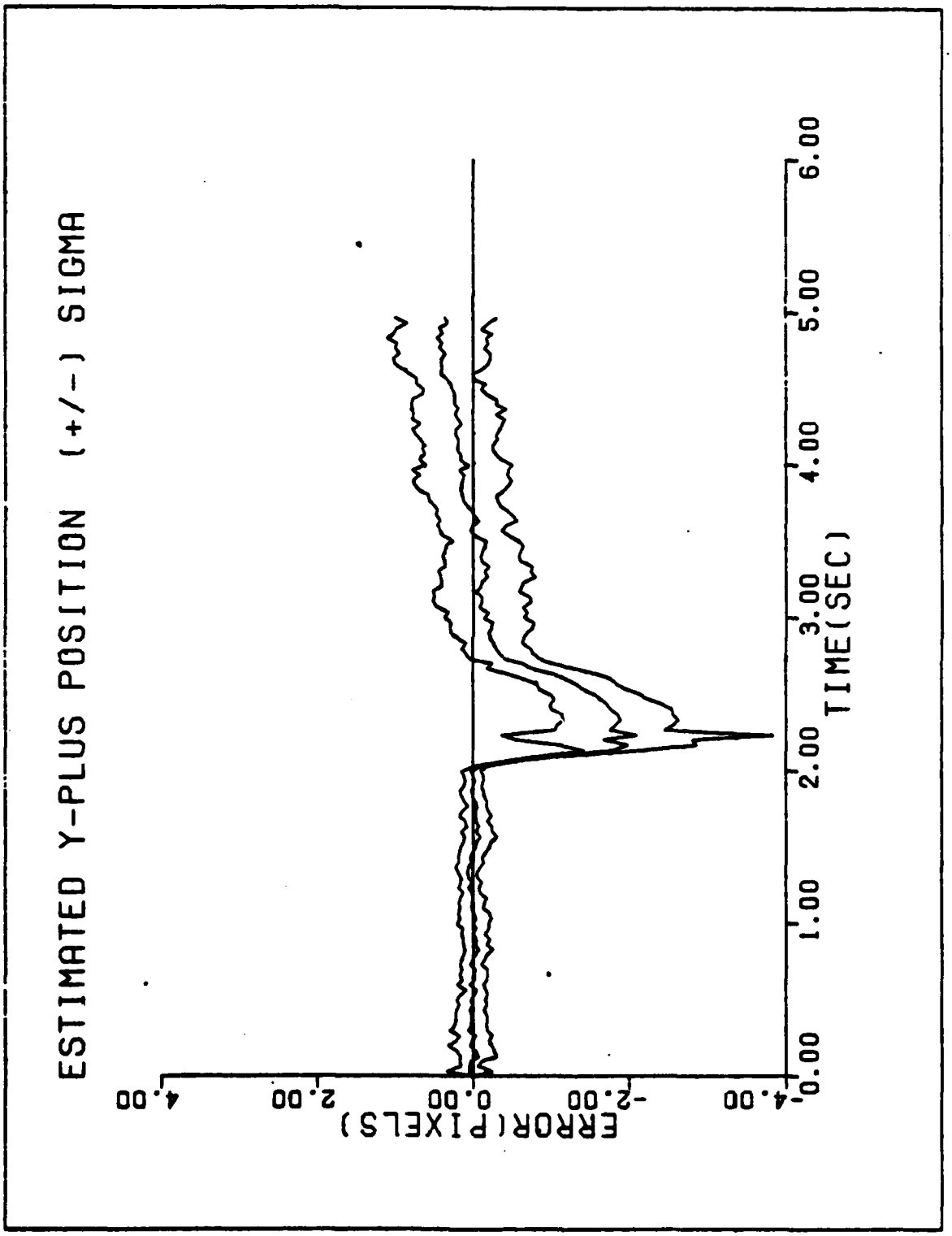


Figure A-23b. Performance Plot for L20T4MF

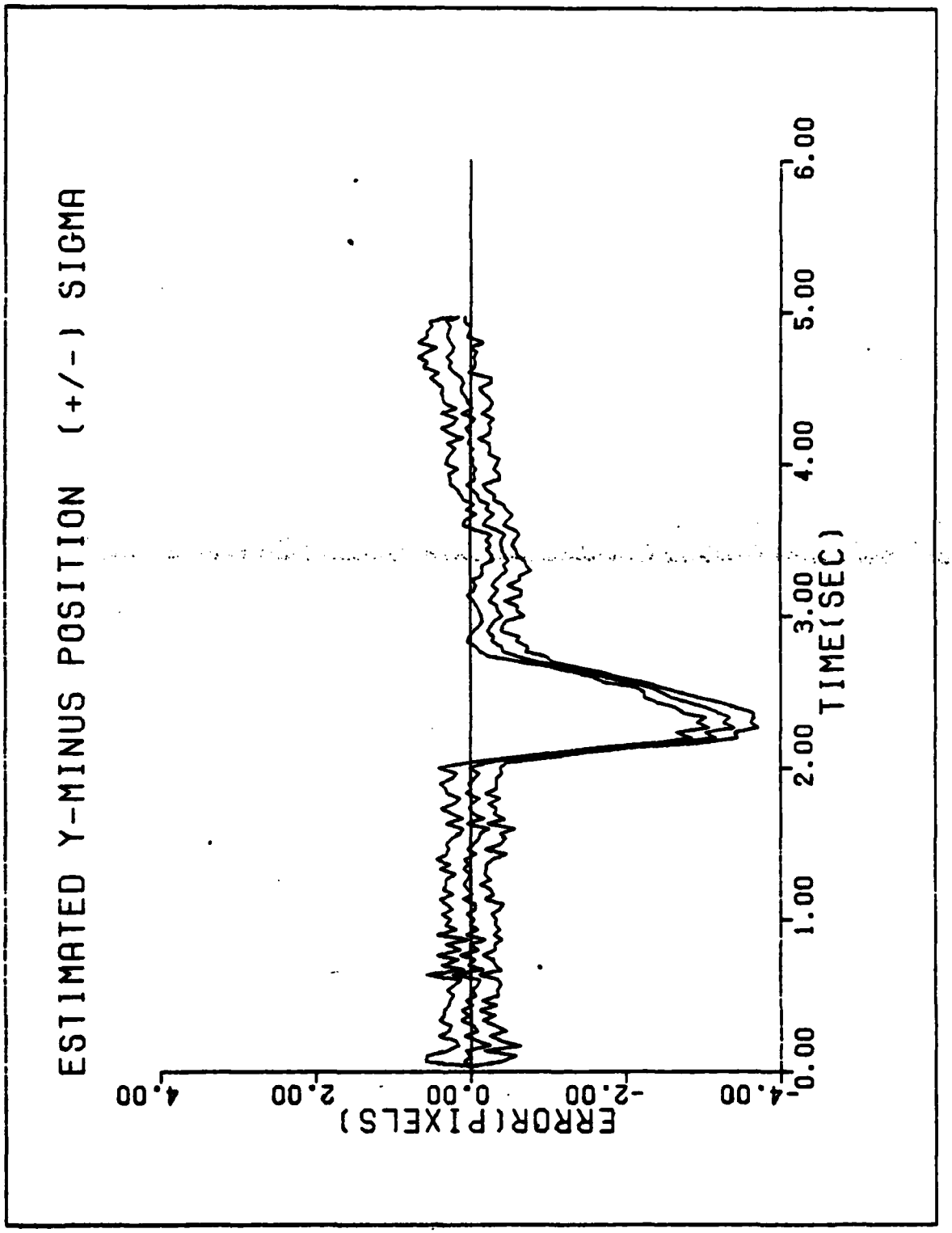


Figure A-24a. Performance Plot for L20T4F2

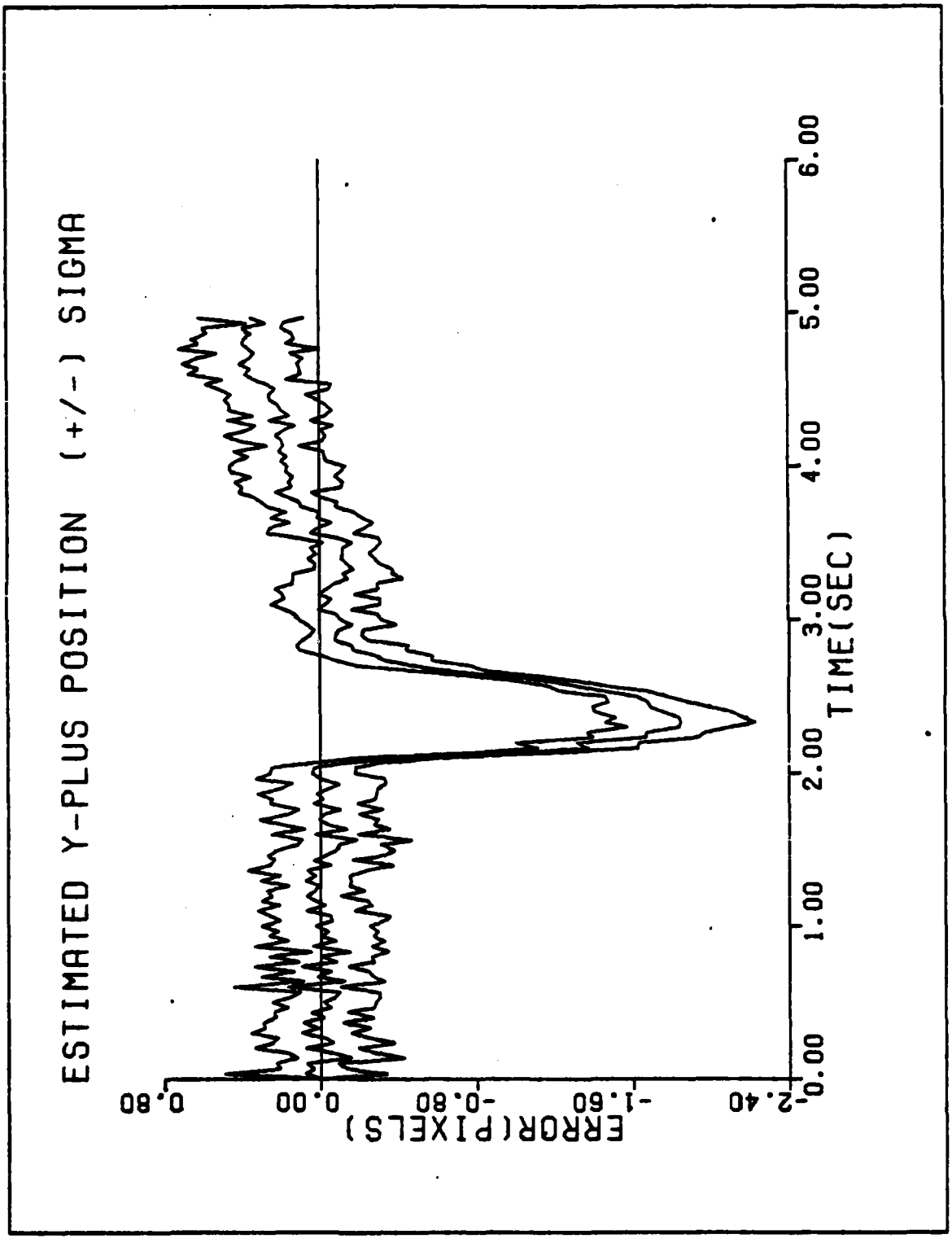


Figure A-24b. Performance Plot for L20T4F2

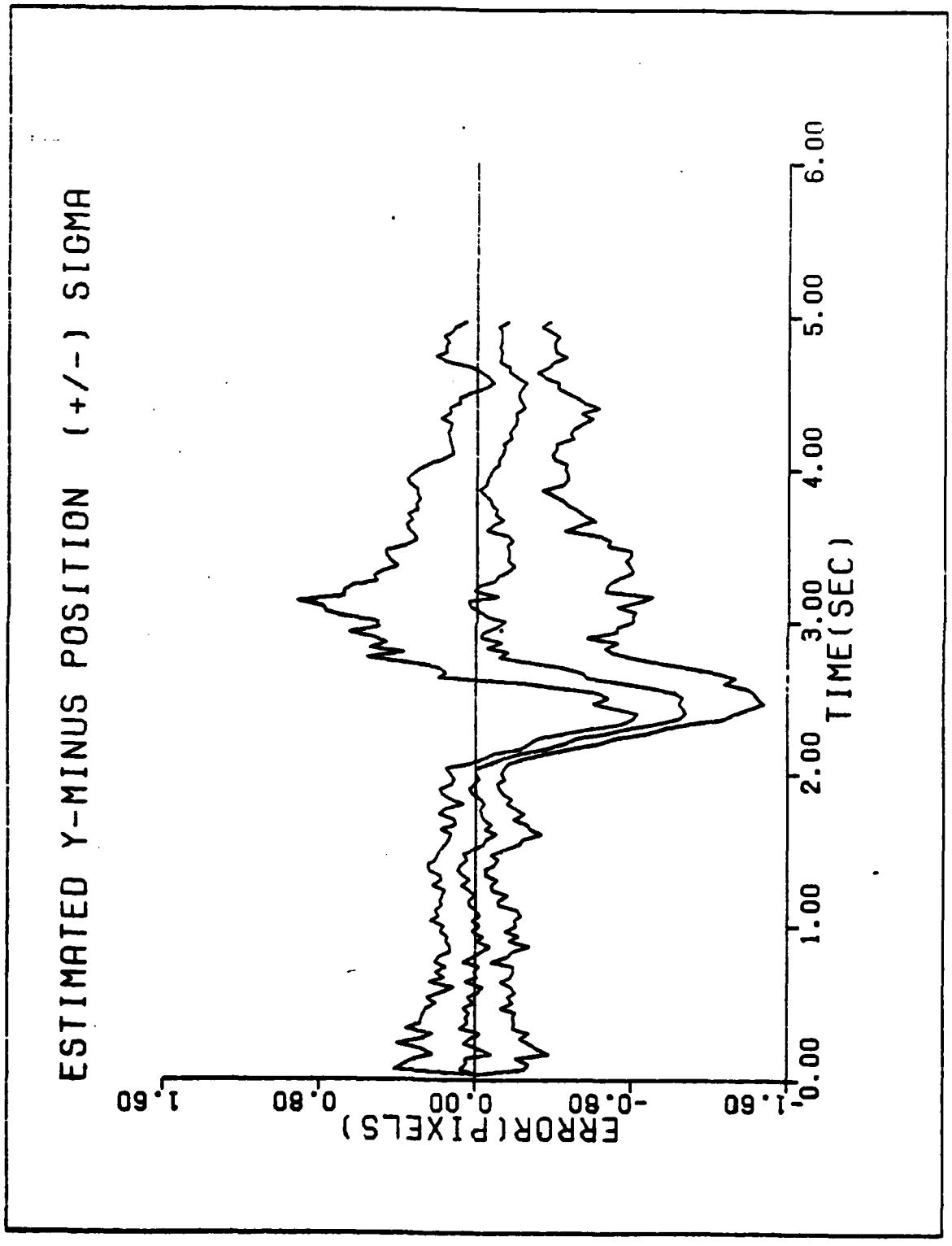


Figure A-25a. Performance Plot for LS2T2MF

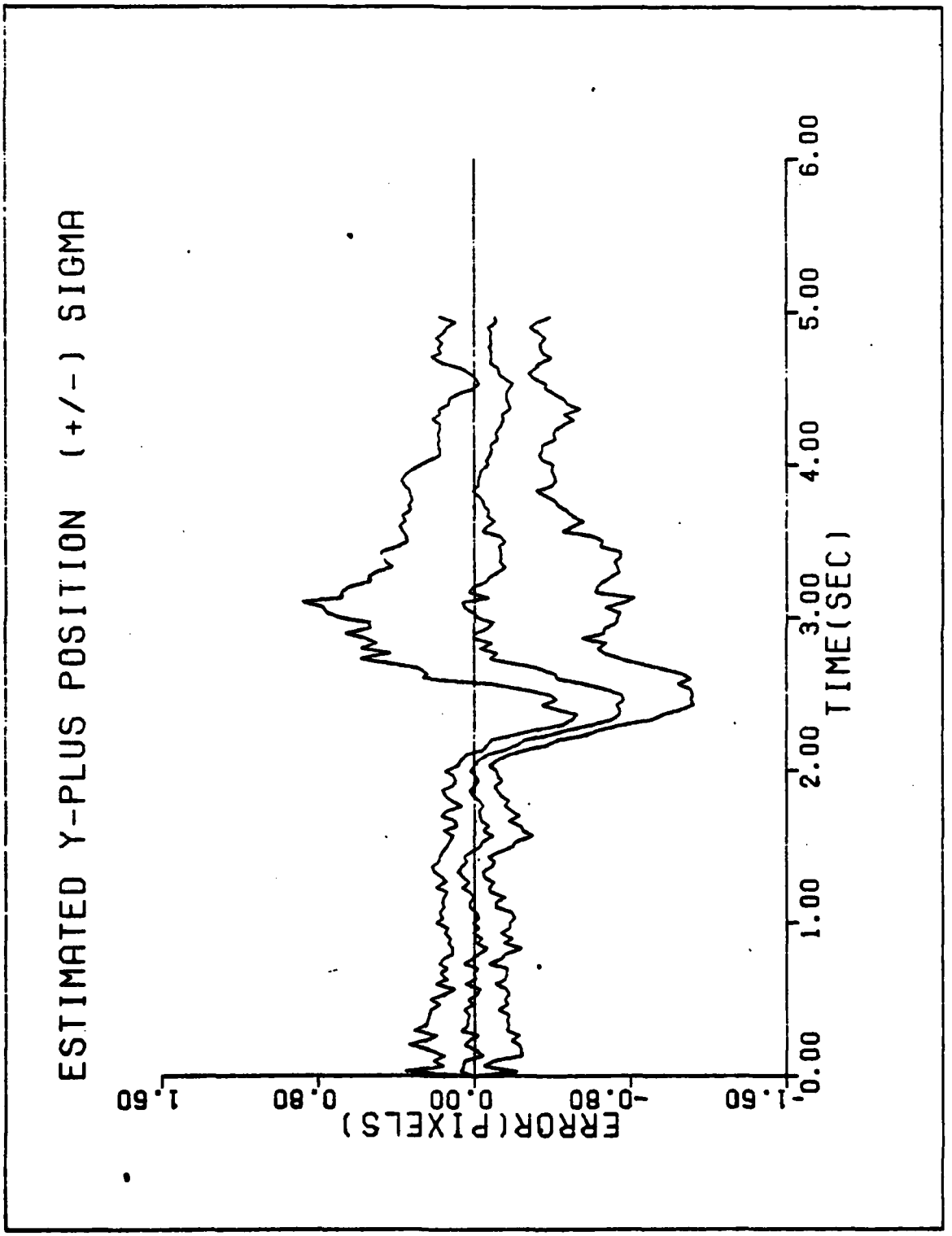


Figure A-25b. Performance Plot for LS2T2MF

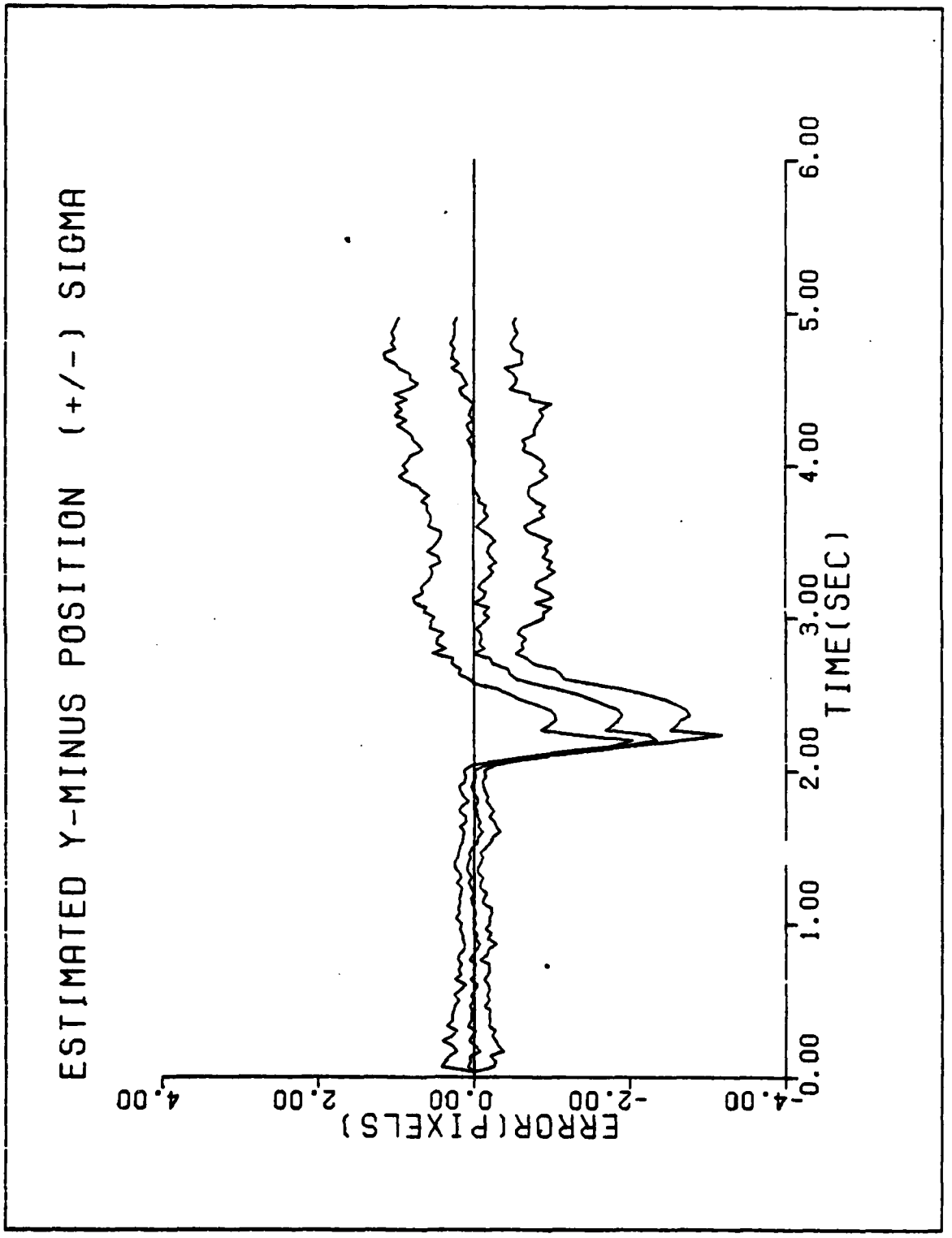


Figure A-26a. Performance Plot for LS10T2MF

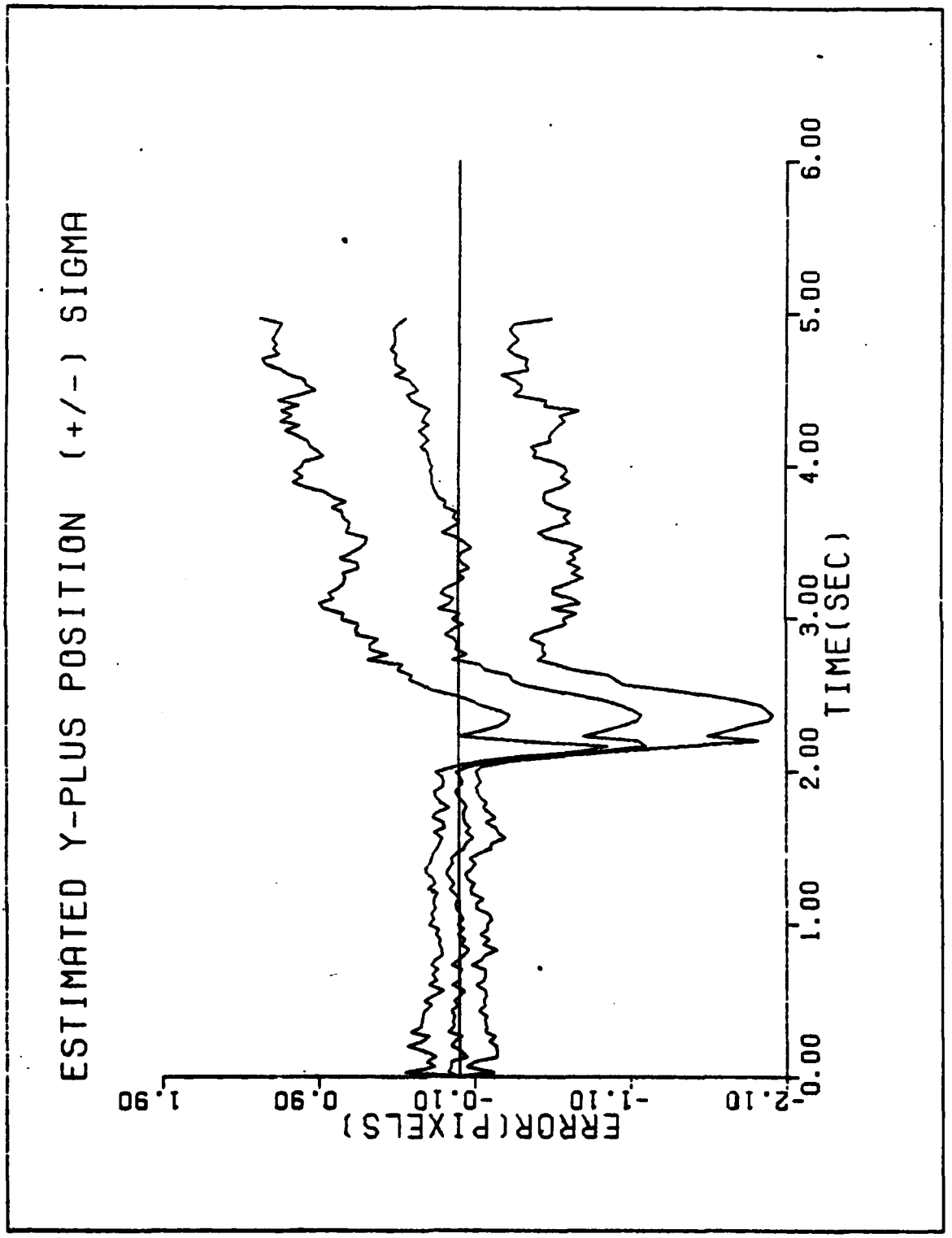


Figure A-26b. Performance Plot for LS10T2MF

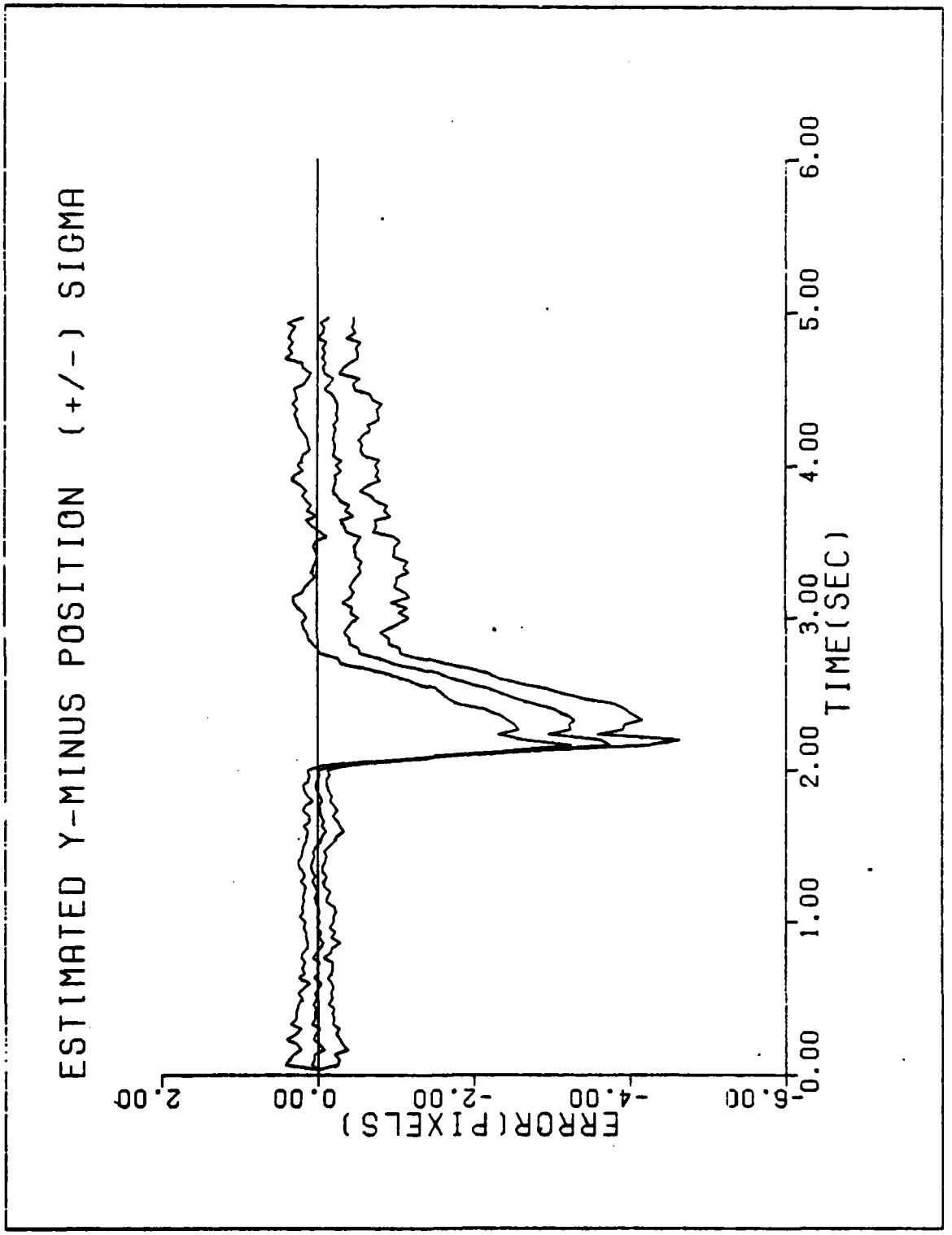


Figure A-27a. Performance Plot for LS20T2MF

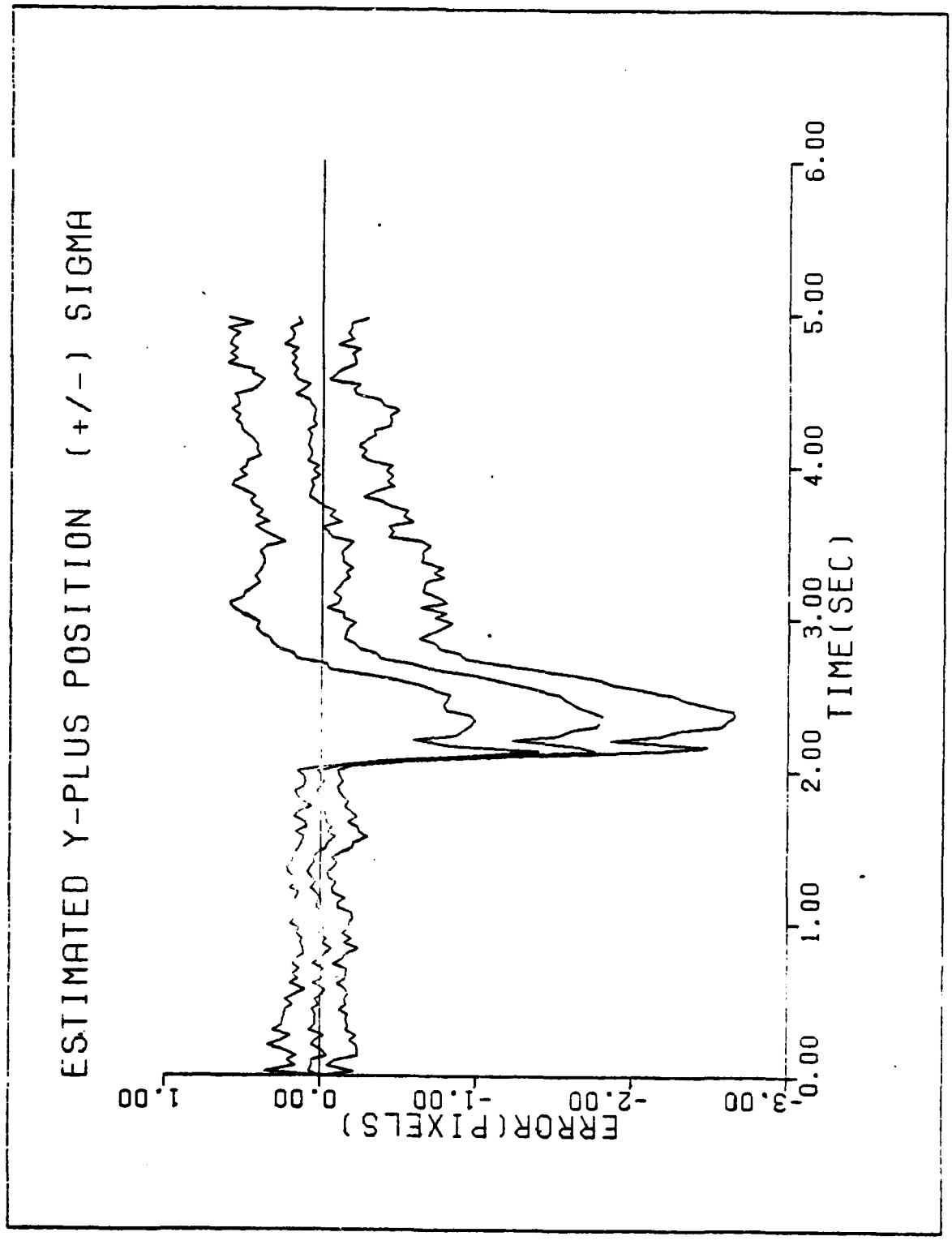


Figure A-27b. Performance Plot for LS20T2MF

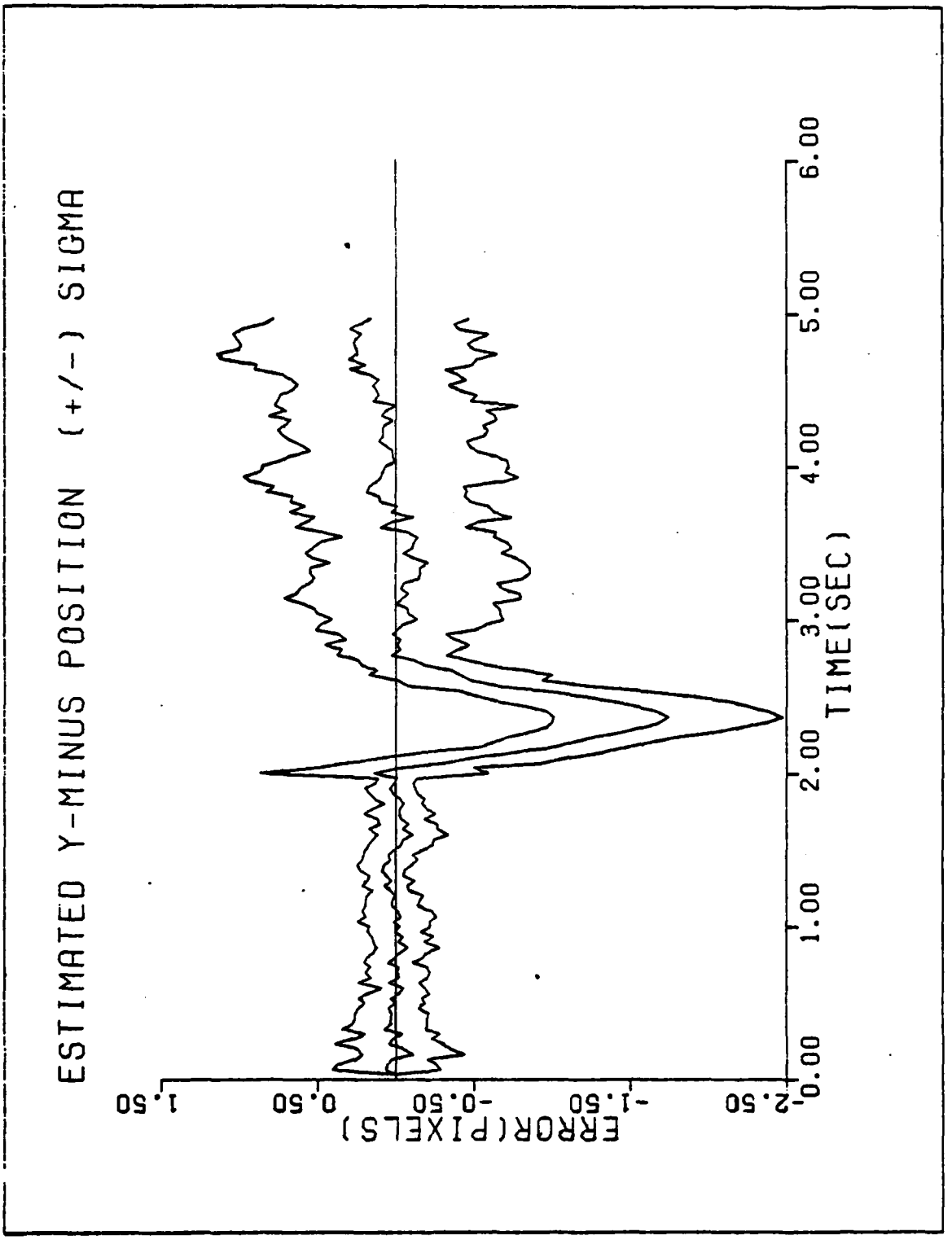


Figure A-28a. Performance Plot for LA10T2MF

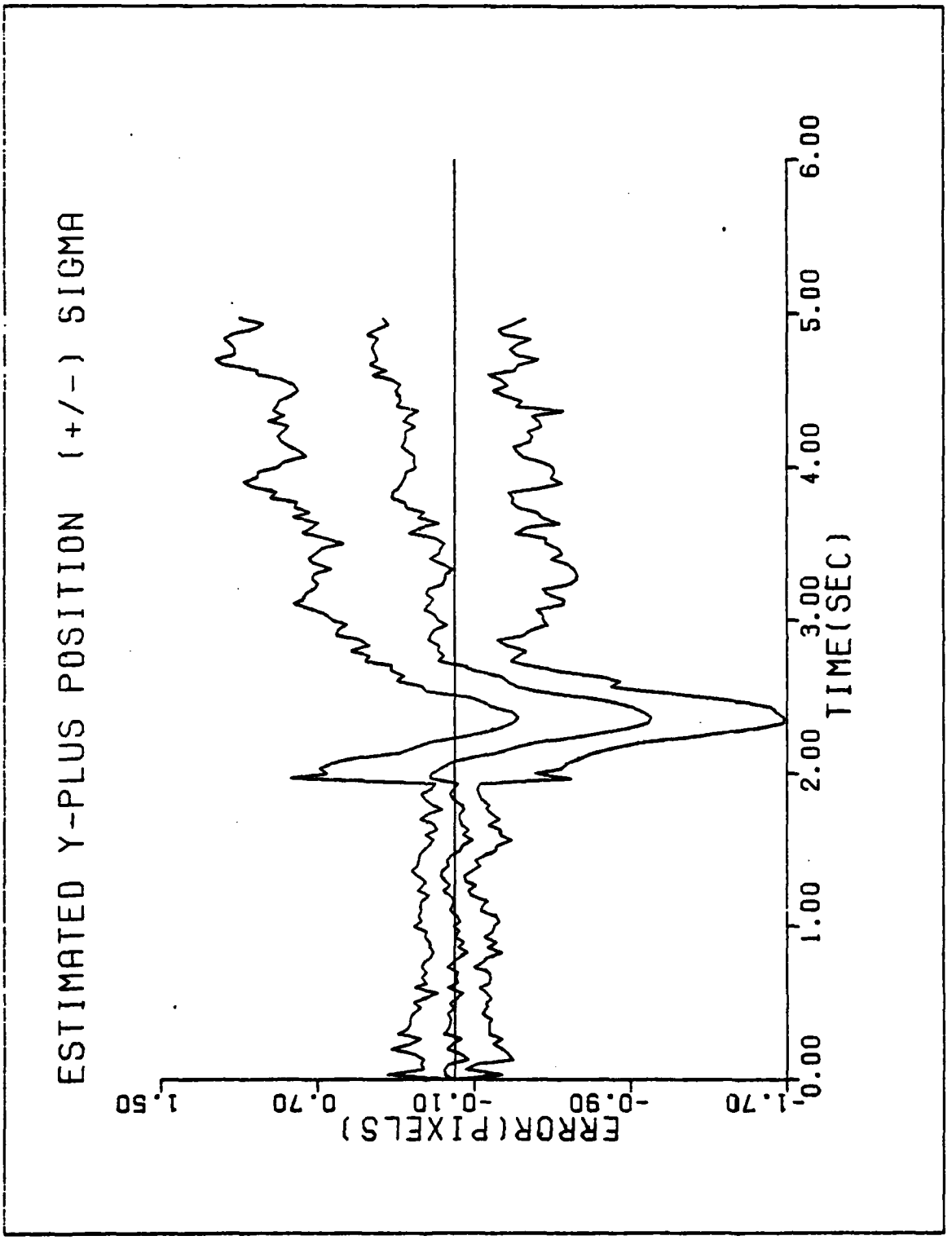


Figure A-28b. Performance Plot for LA10T2MF

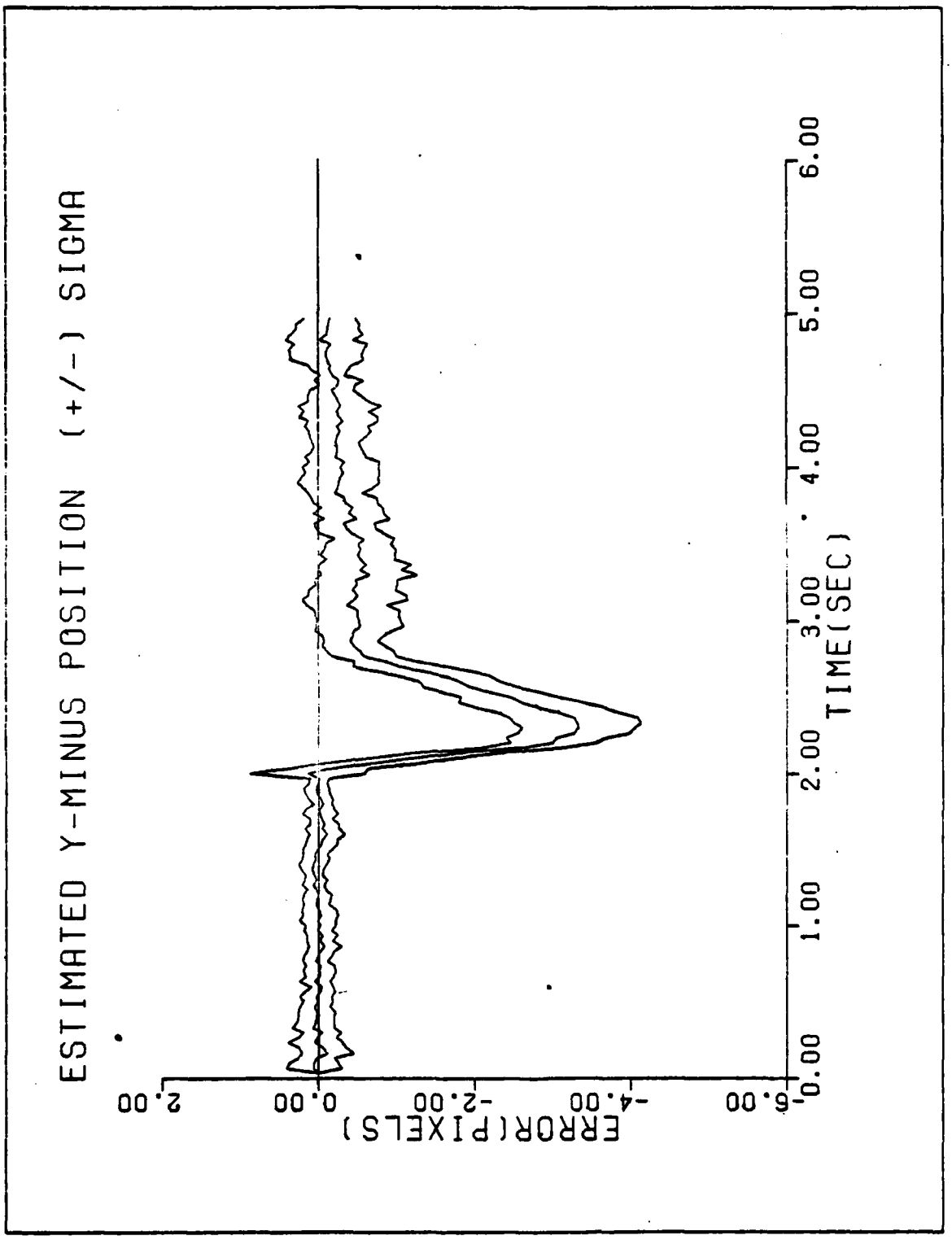


Figure A-29a. Performance Plot for LA20T2MF

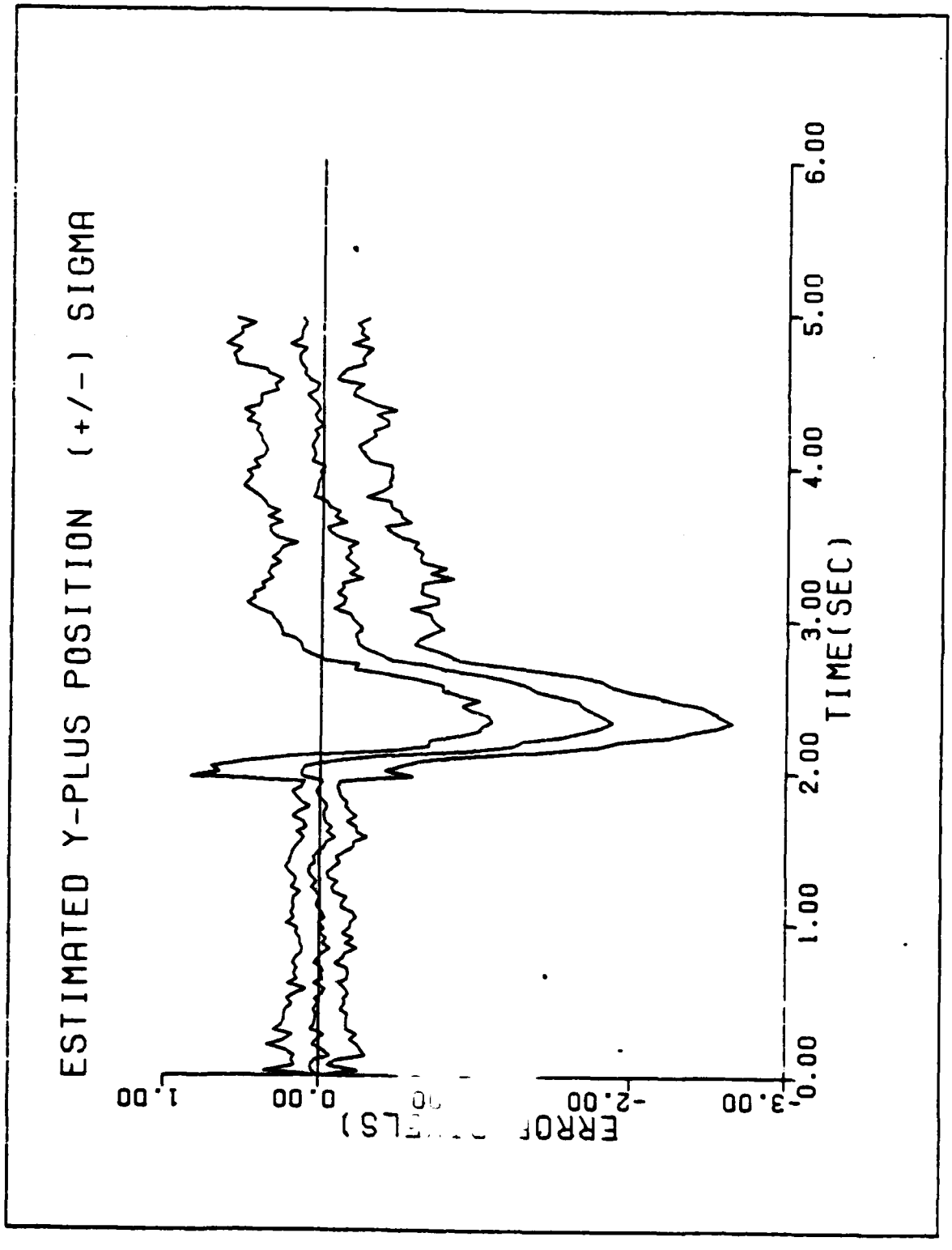


Figure A-29b. Performance Plot for LA20T2MF

Appendix B

This appendix contains the performance plots for the extended Kalman filter tracker configuration using a Gauss-Markov target acceleration model. Thes cases are identified according to the mnemonic code described in Figure V-3. The values for the parameters of these simulations are also found in Chapter V.

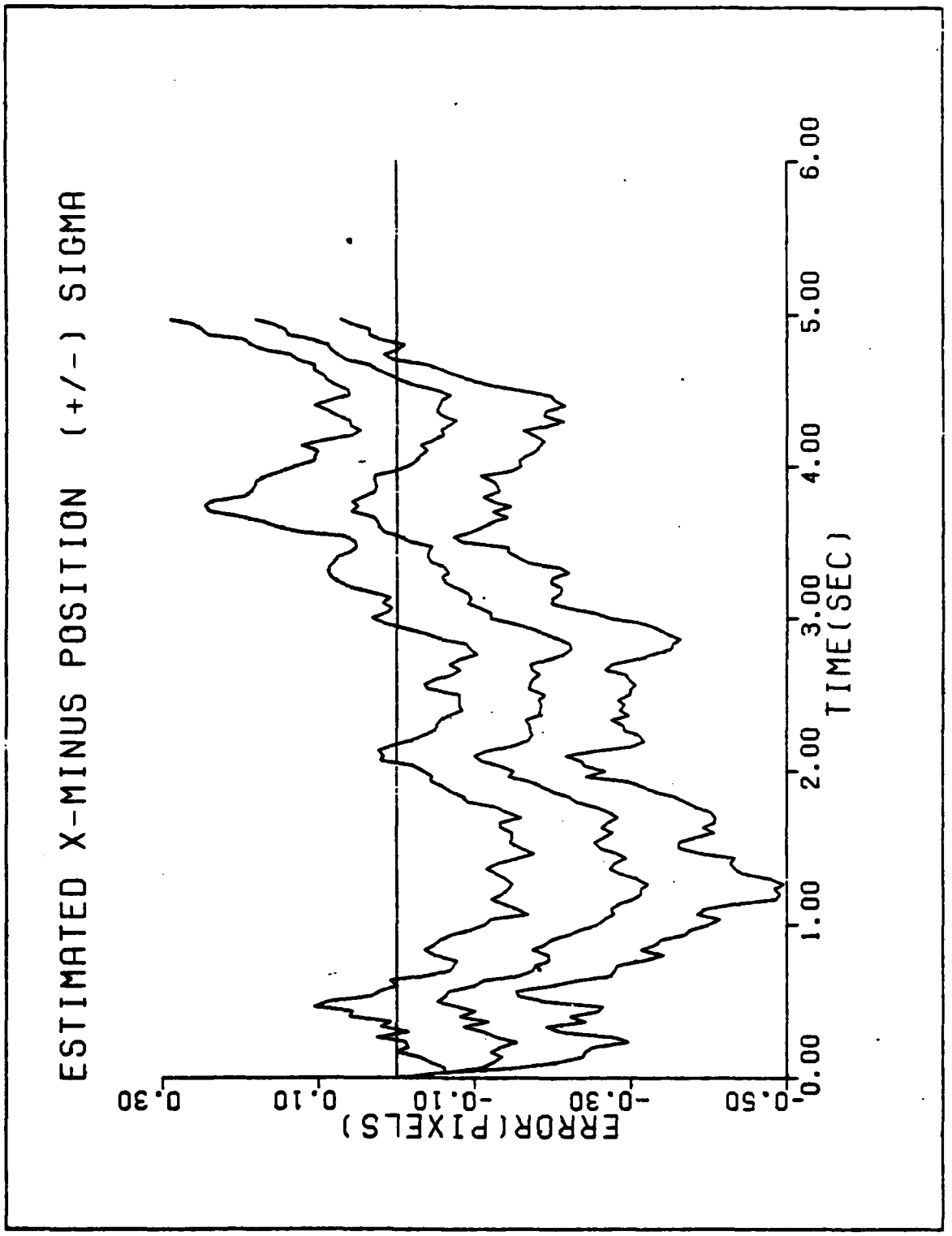


Figure B-1a. Performance Plot for GT1MF

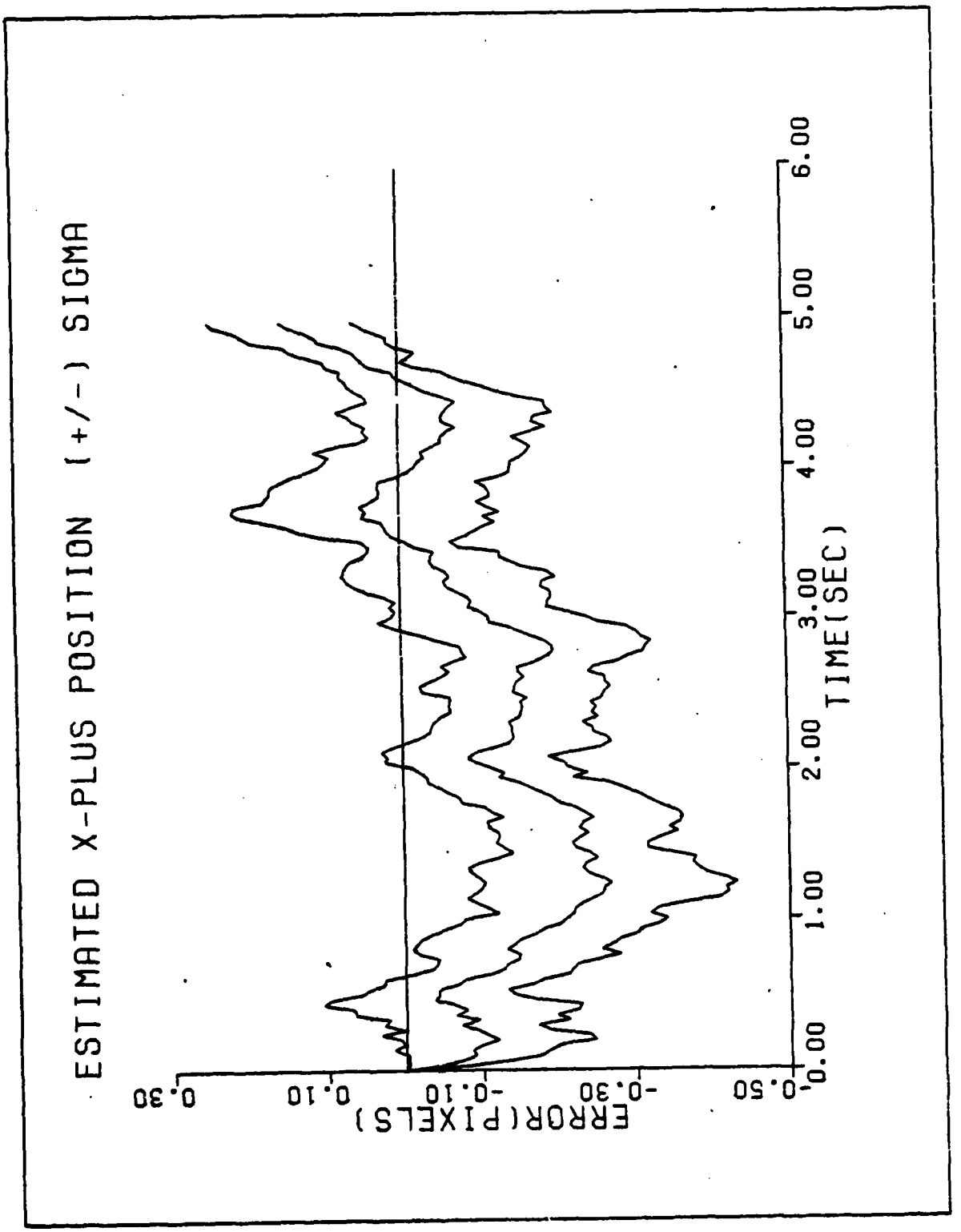


Figure B-1b. Performance Plot for GT1MF

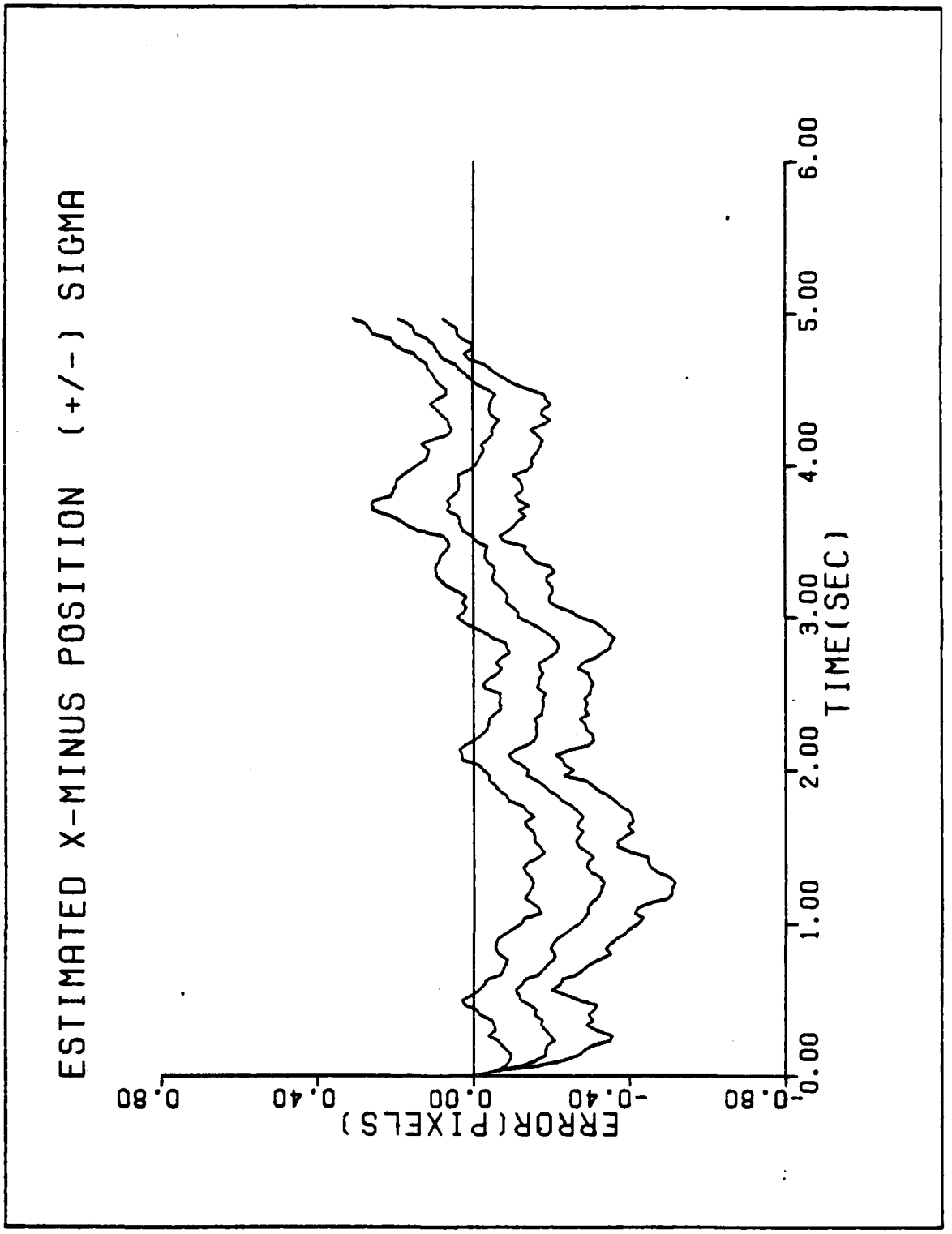


Figure B-2a. Performance Plot for GT1F1

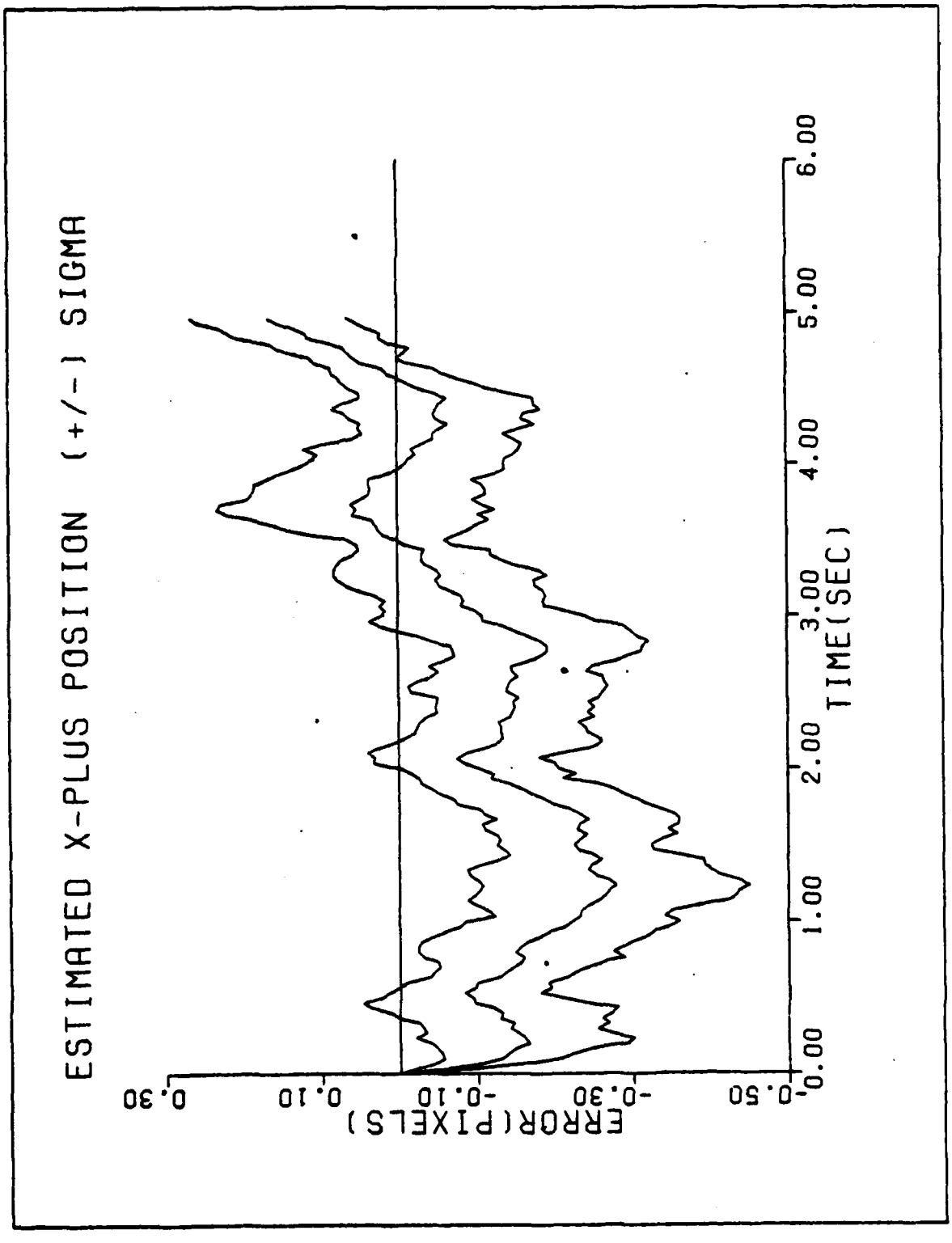


Figure B-2b. Performance Plot for GT1F1

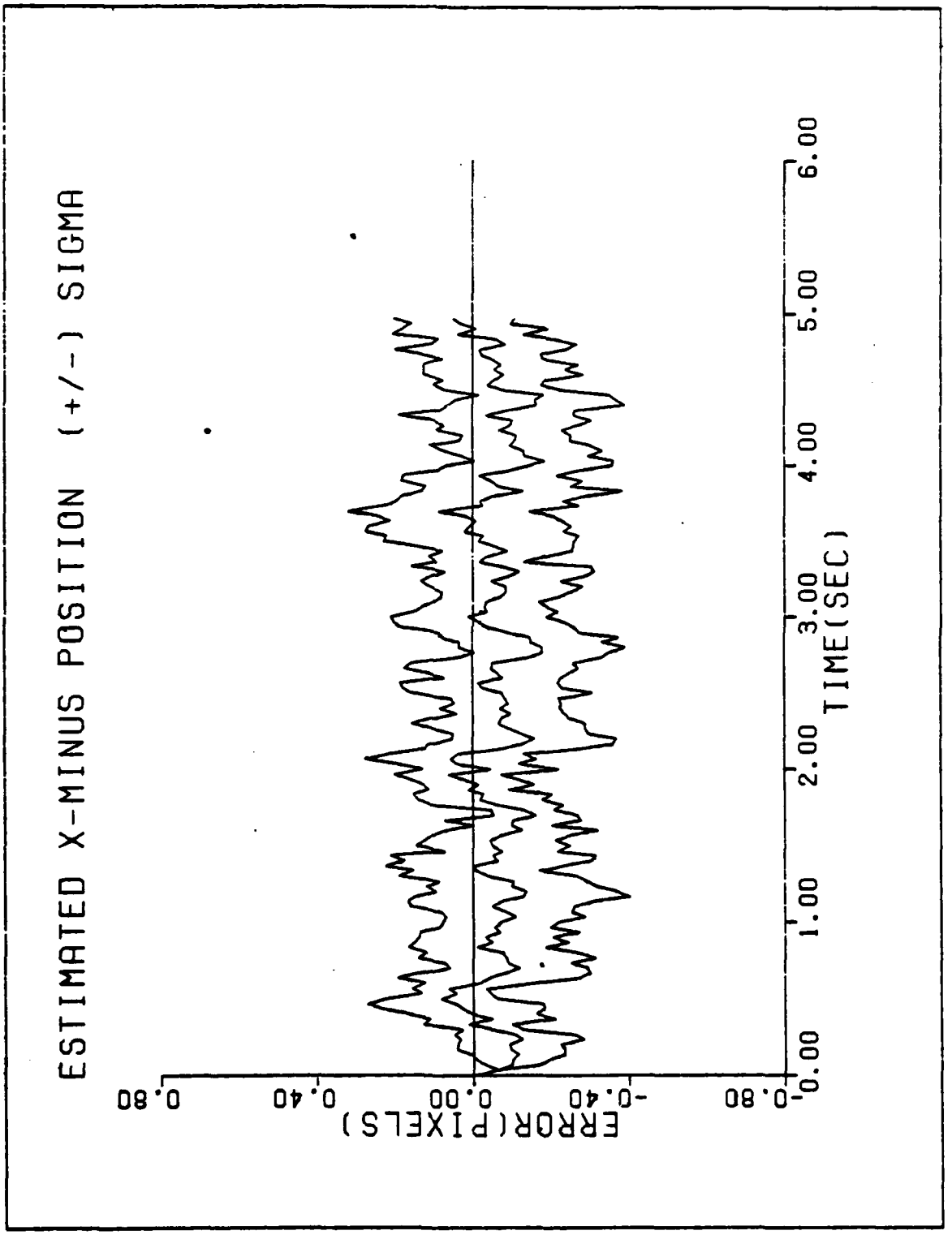


Figure B-3a. Performance Plot for GT1F2

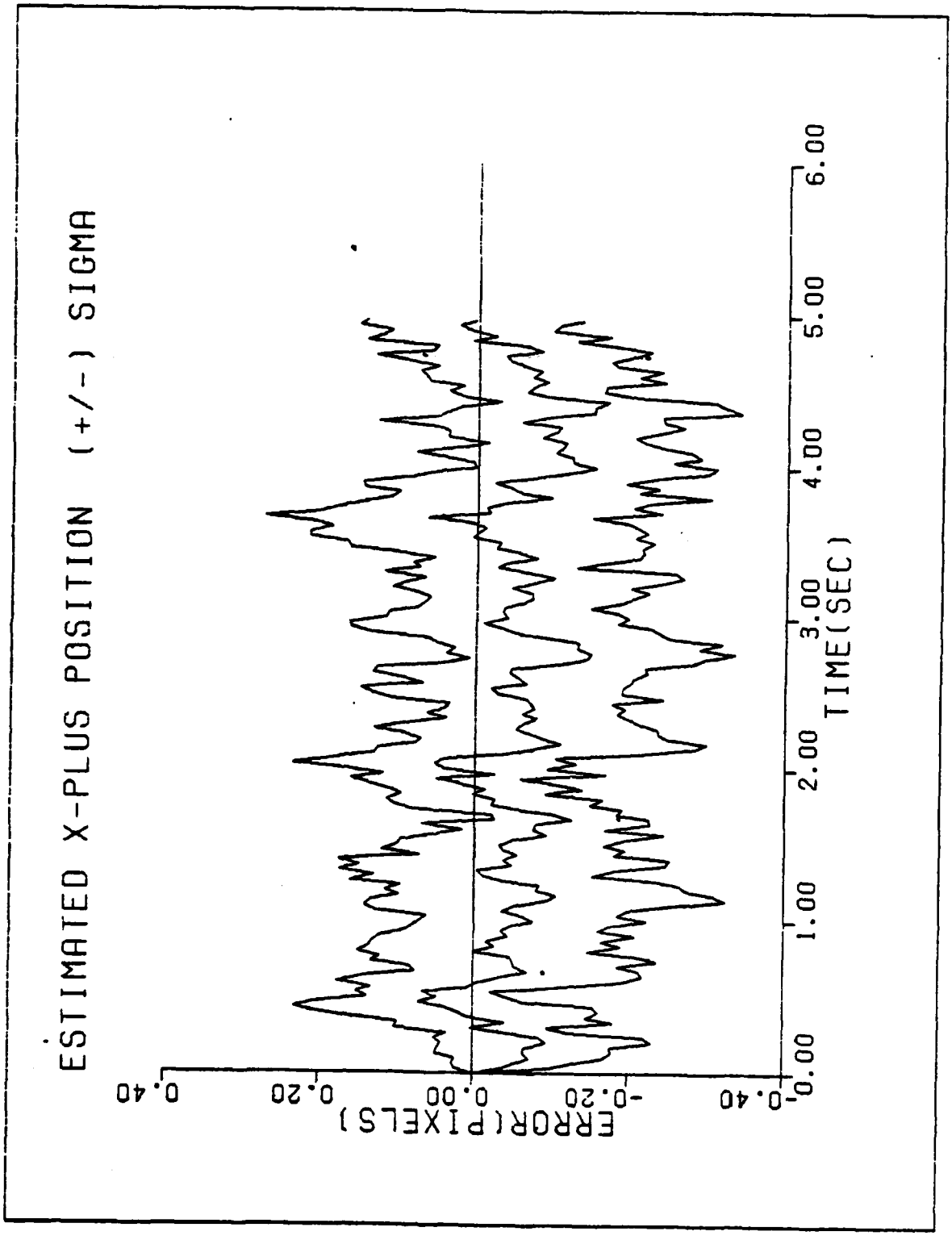


Figure B-3b. Performance Plot for GT1F2

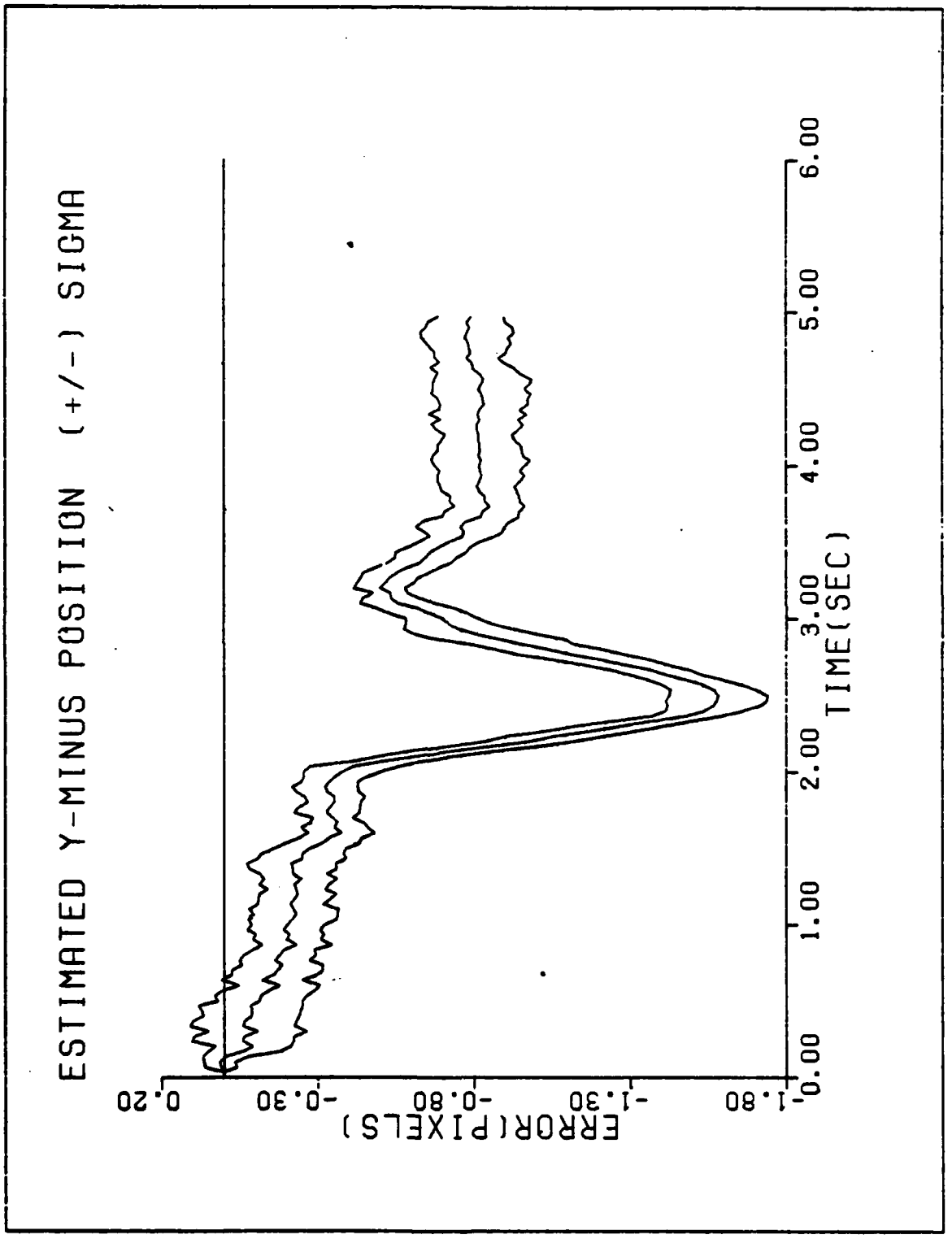


Figure B-4a. Performance Plot for G2T2MF

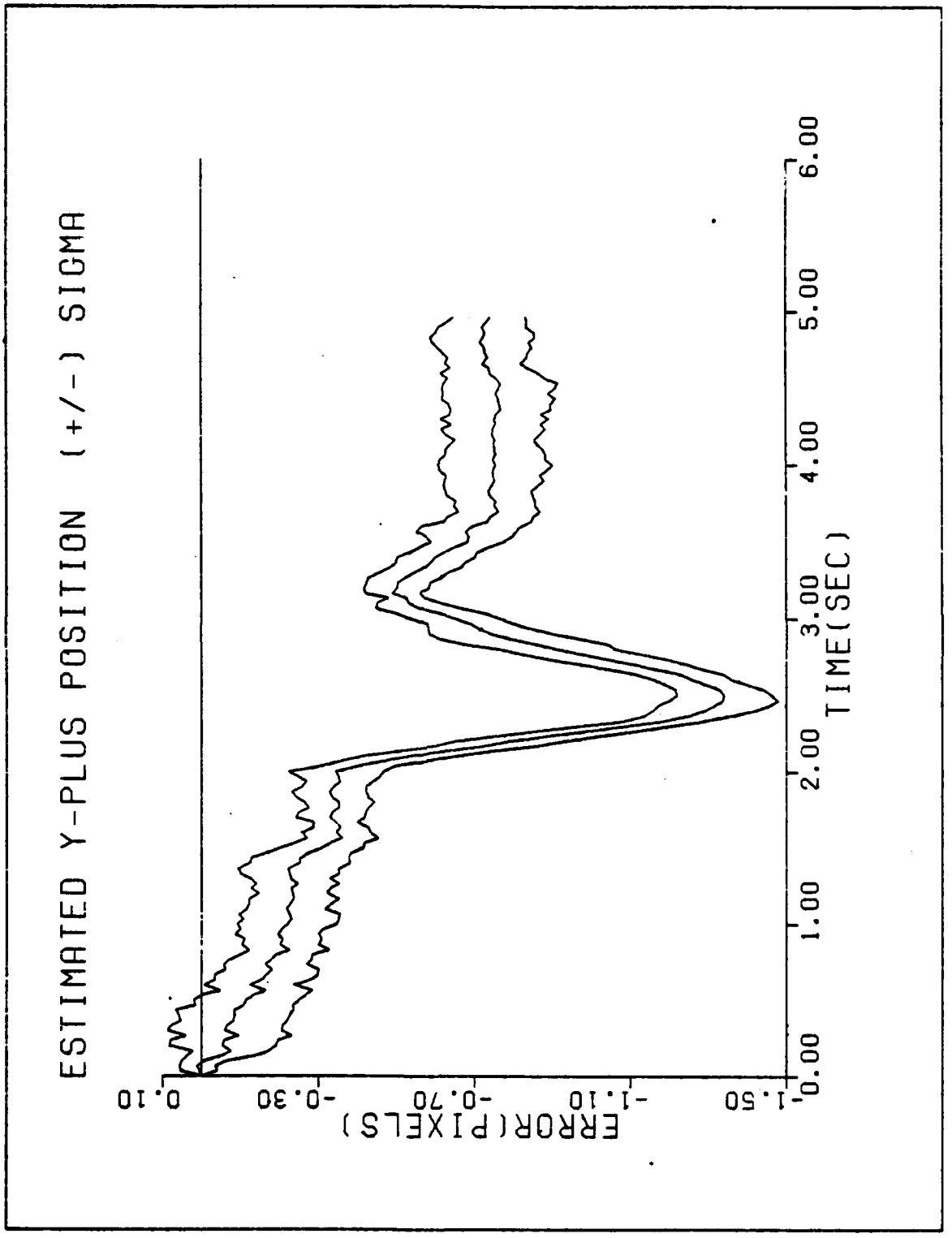


Figure B-4b. Performance Plot for G2T2MF

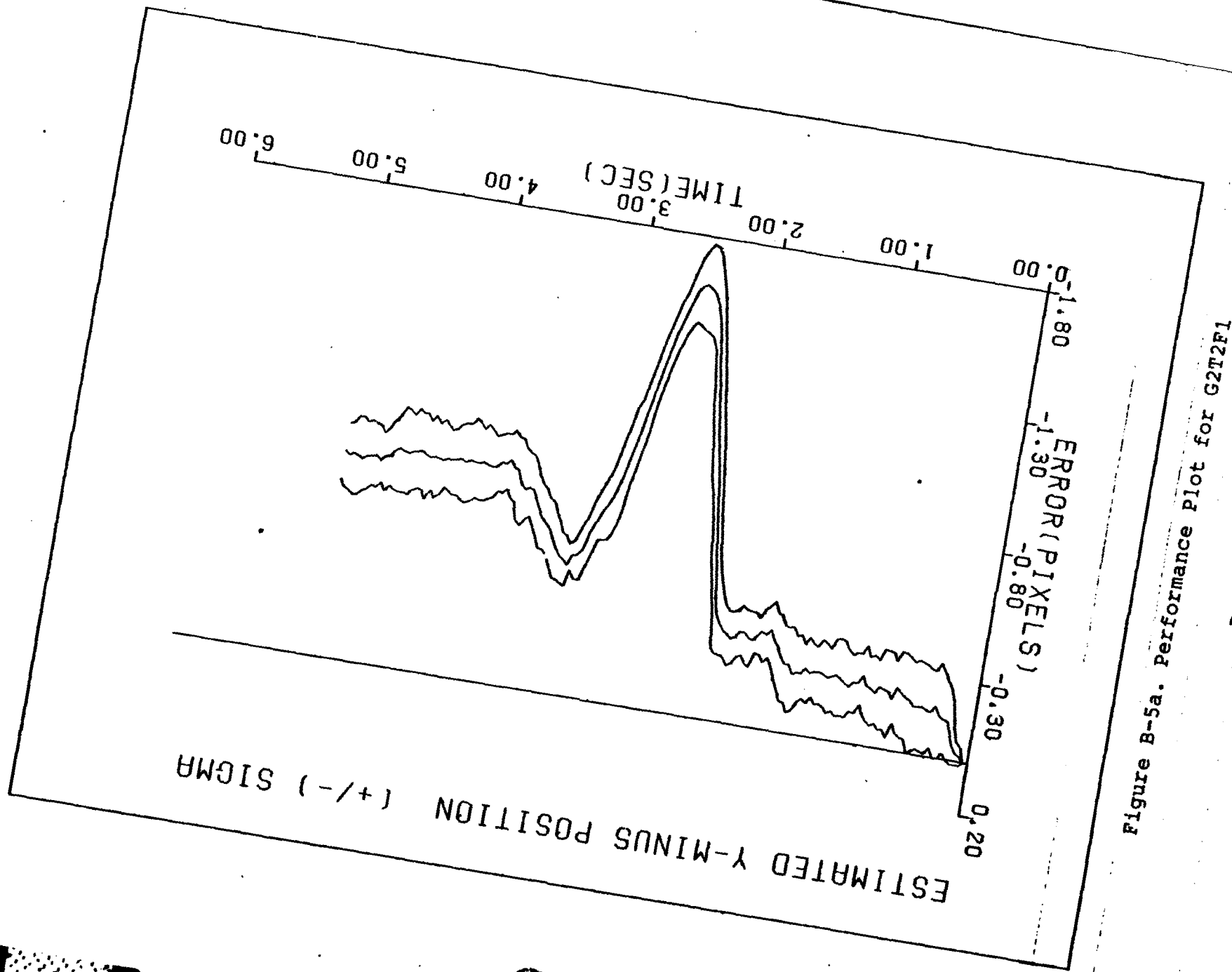


Figure B-5a. Performance Plot for G2T2F1

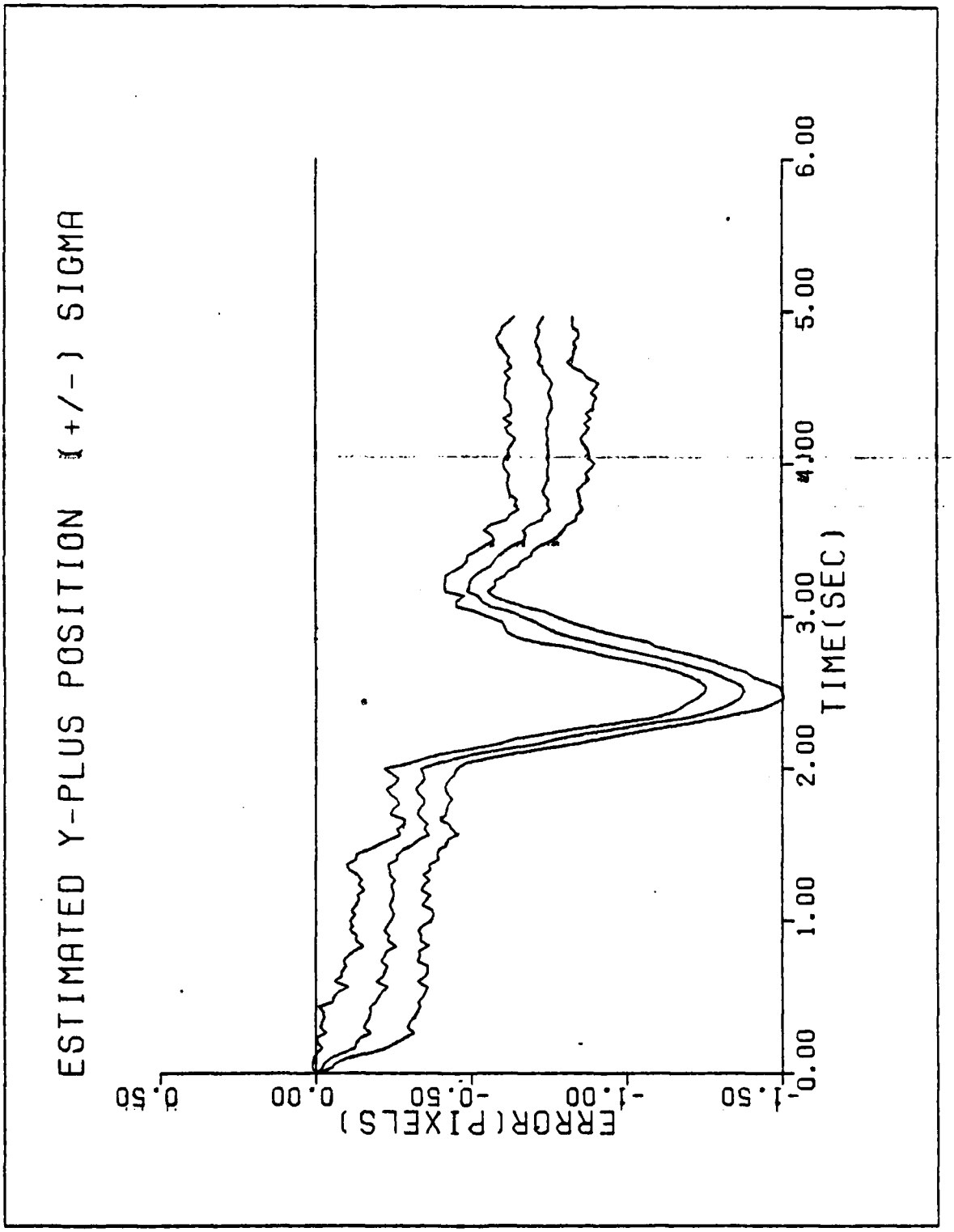


Figure B-5b. Performance Plot for G2T2F1

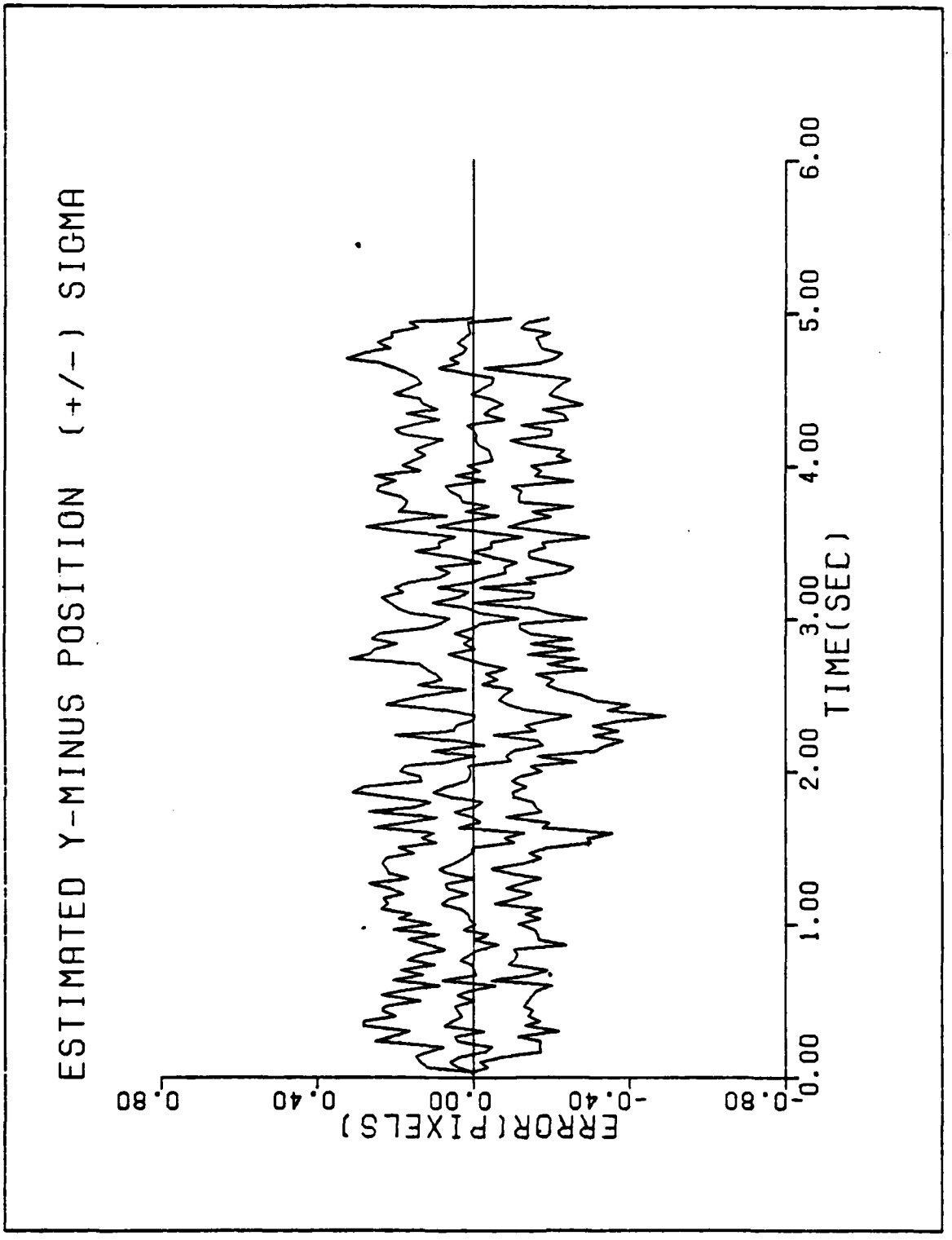


Figure B-6a. Performance Plot for G2T2F2

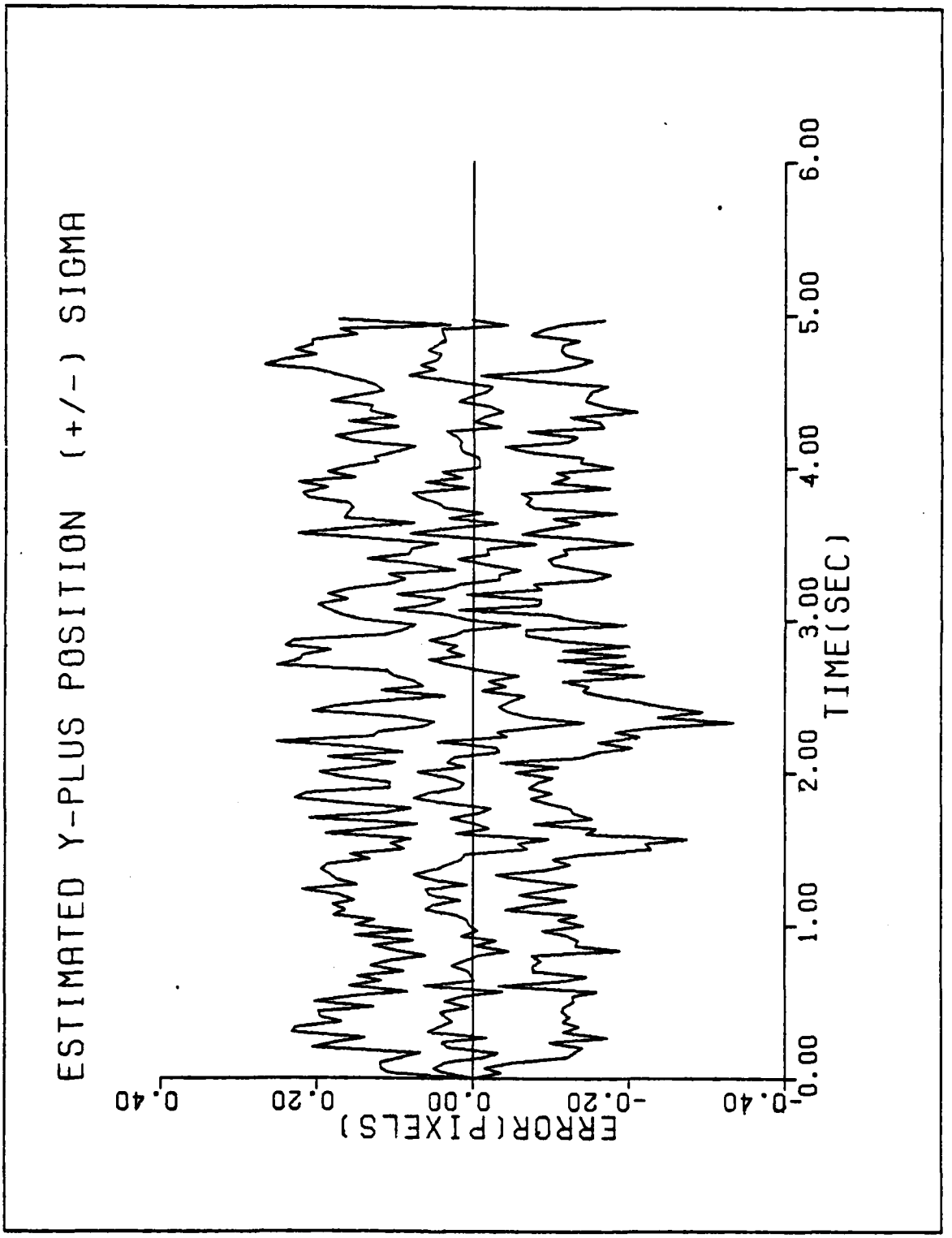


Figure B-6b. Performance Plot for G2T2F2

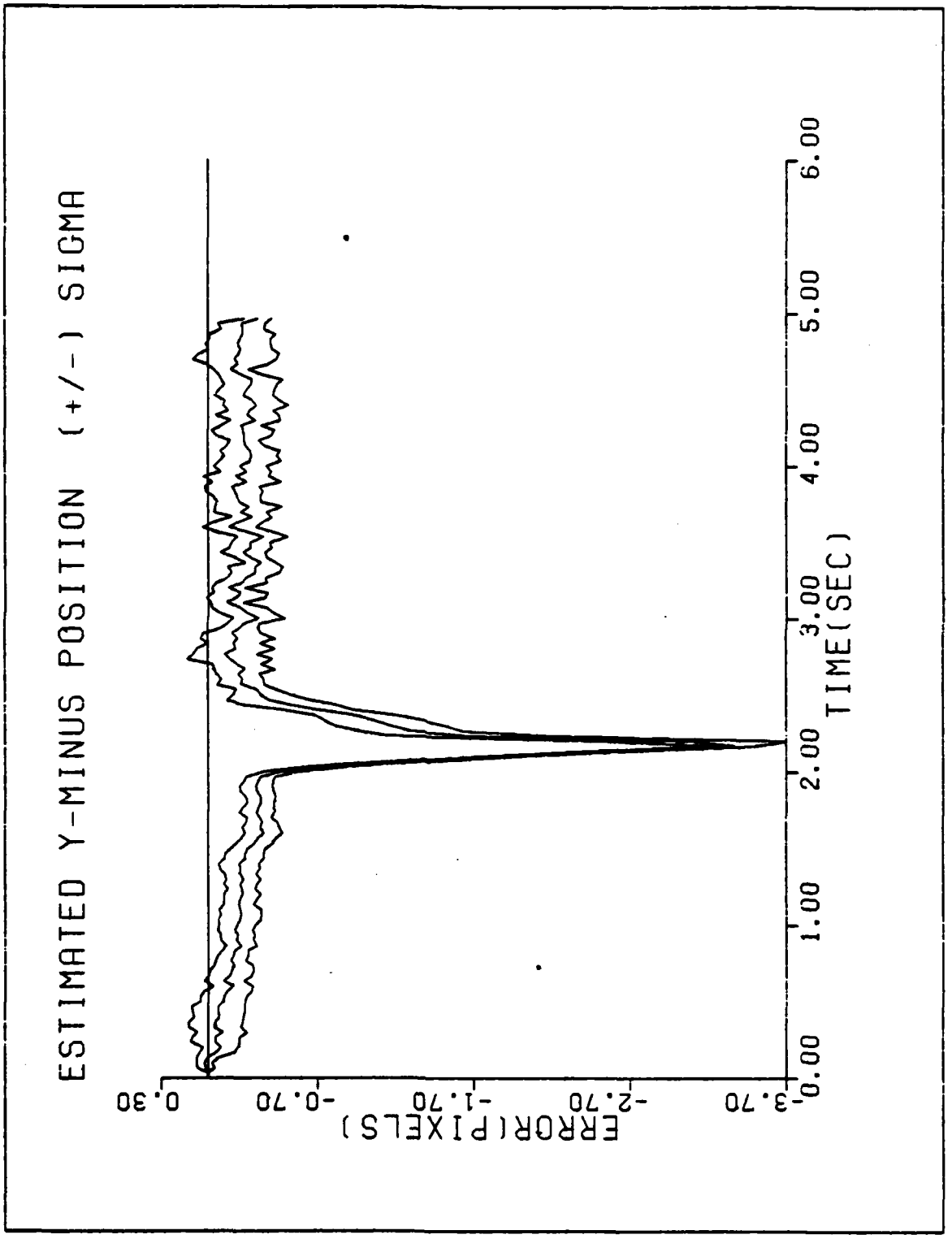


Figure B-7a. Performance Plot for G10T2MF

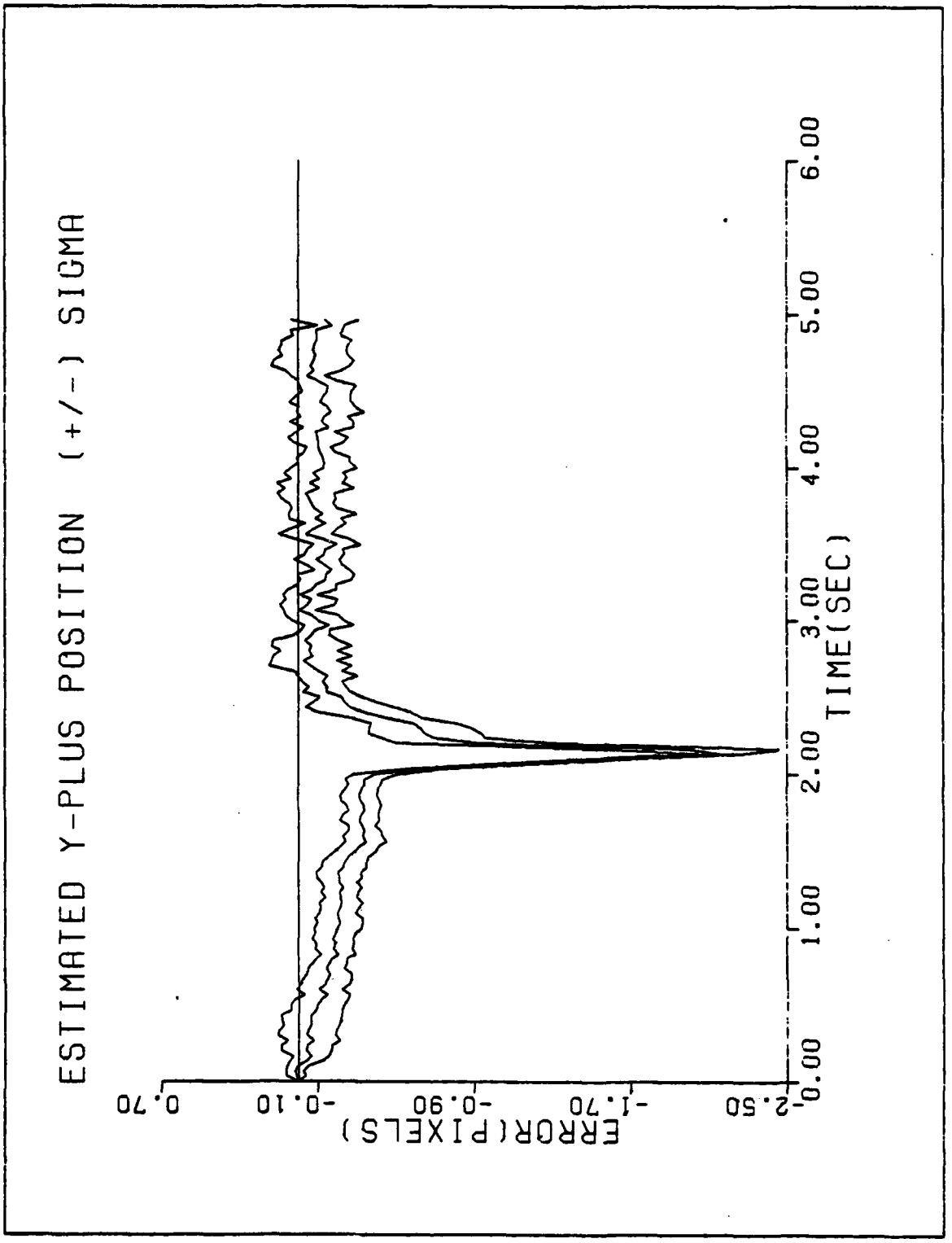


Figure B-7b. Performance Plot for G10T2MF

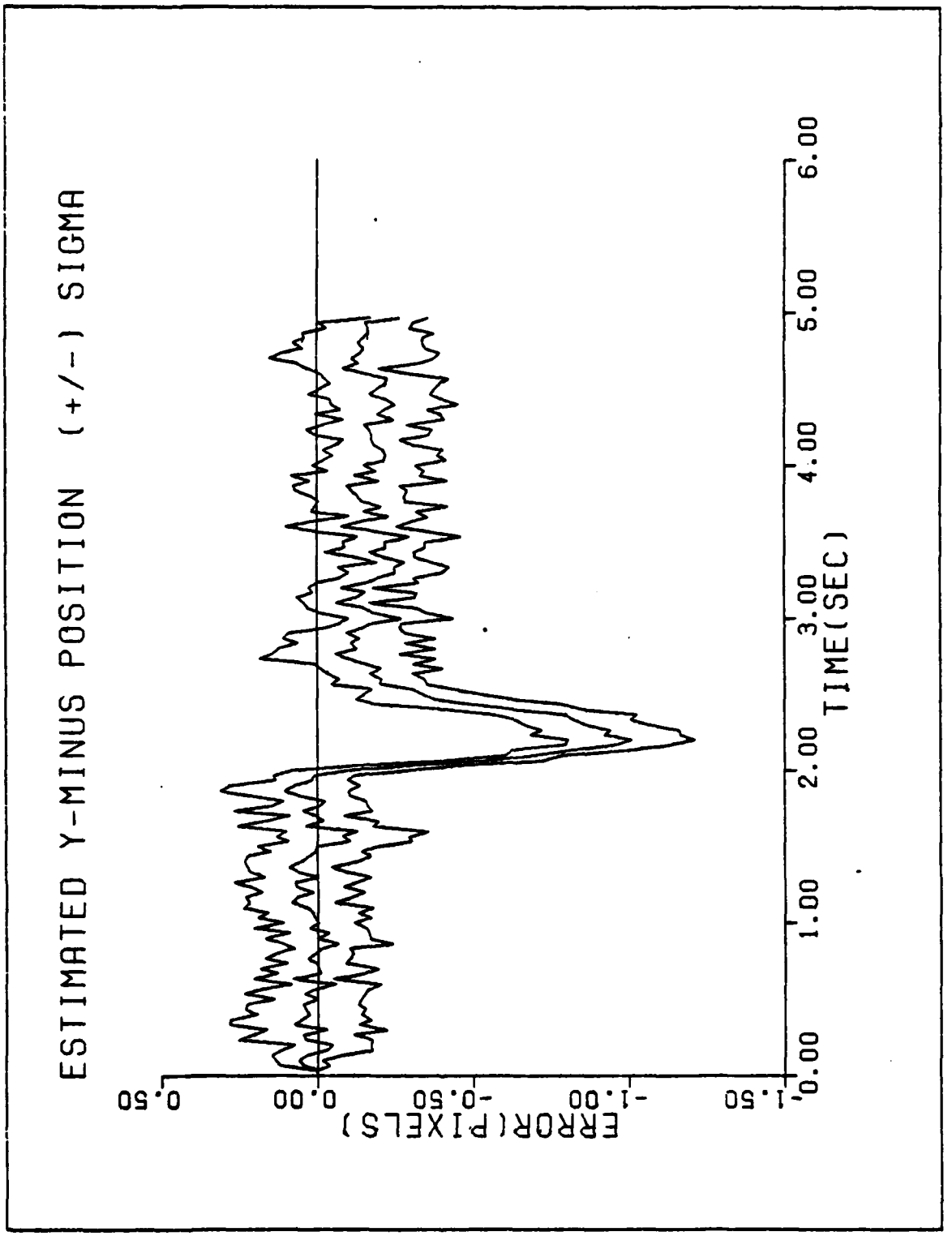


Figure B-8a. Performance Plot for G10T2F2

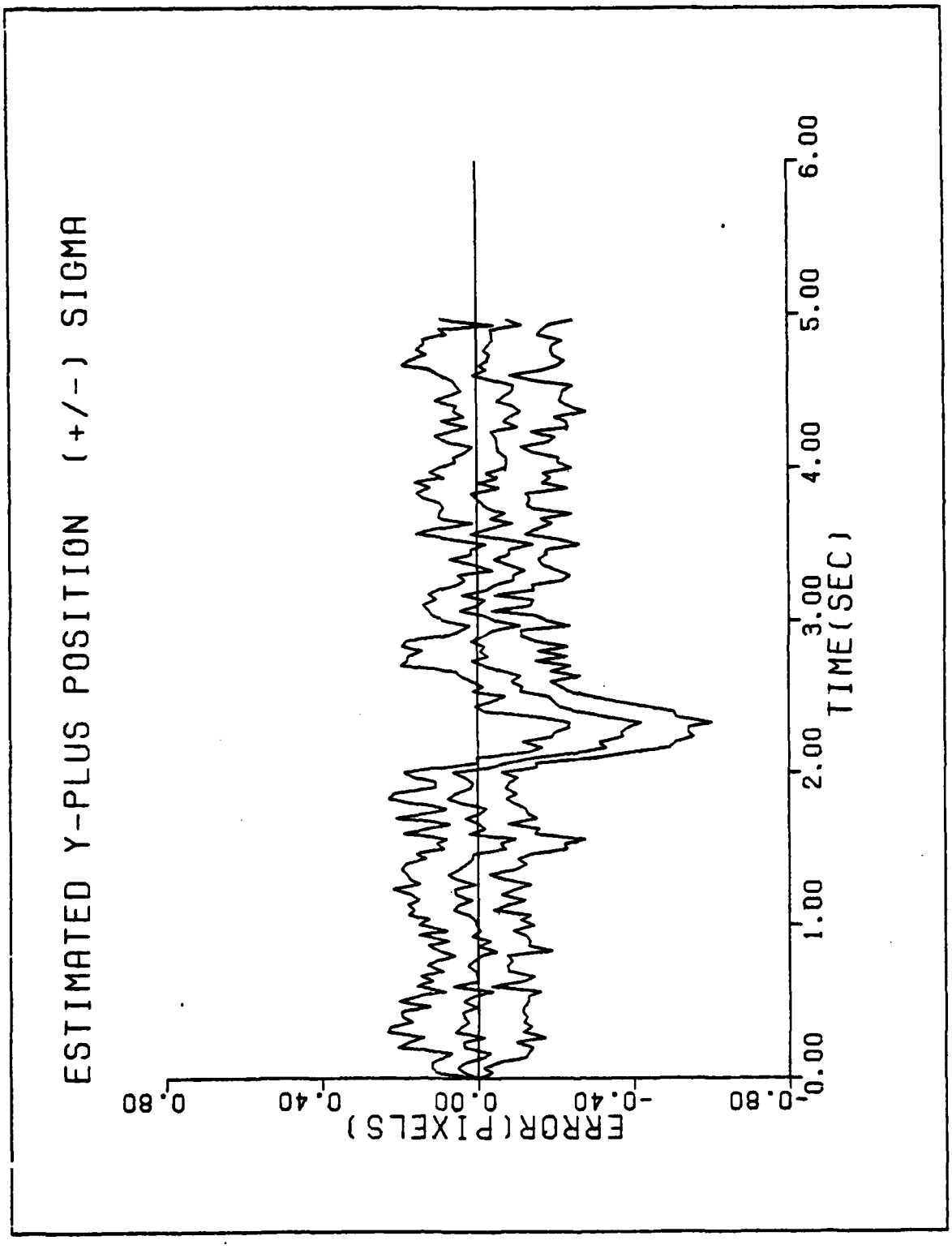


Figure B-8b. Performance Plot for G10T2F2

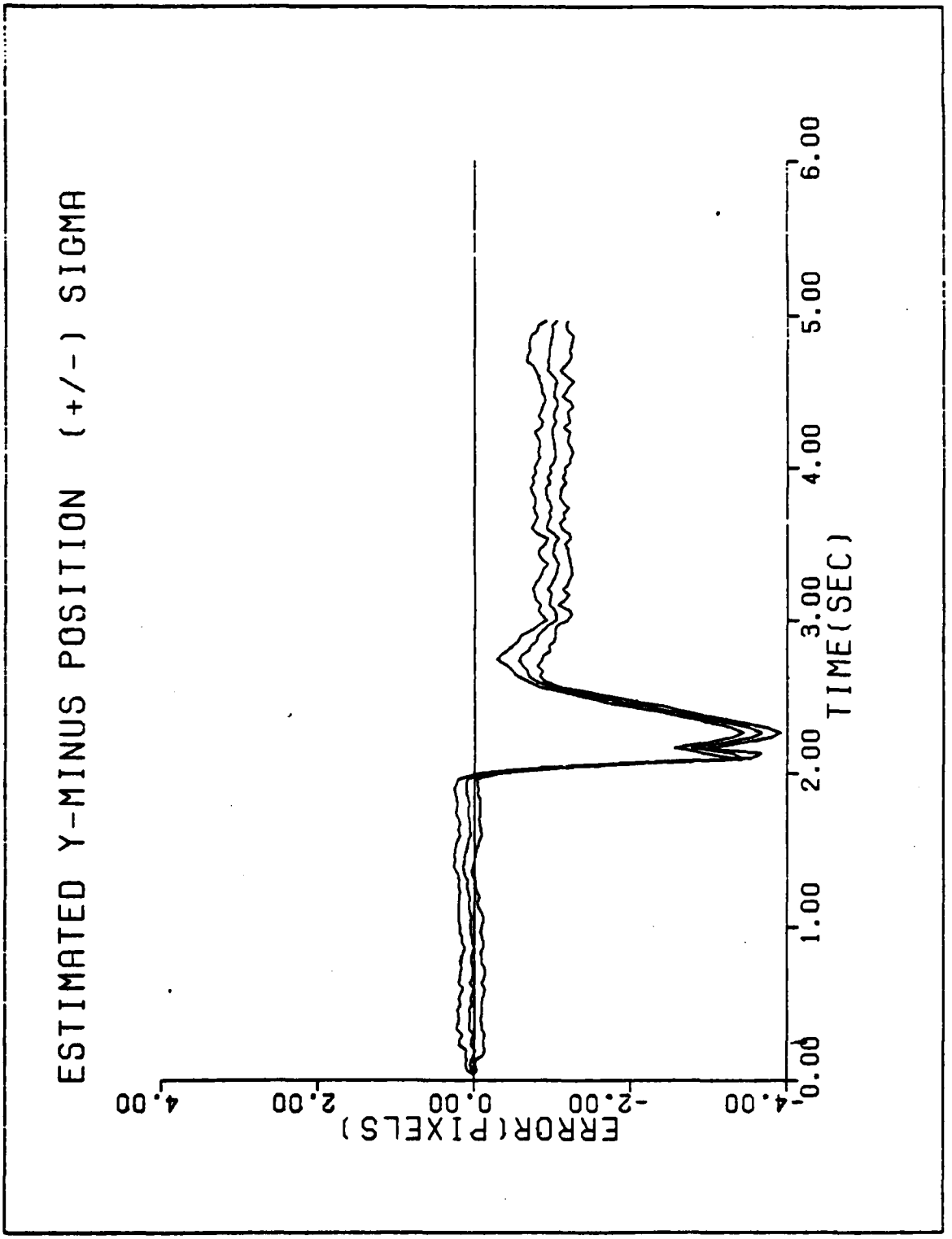


Figure B-9a. Performance Plot for G20T2MF

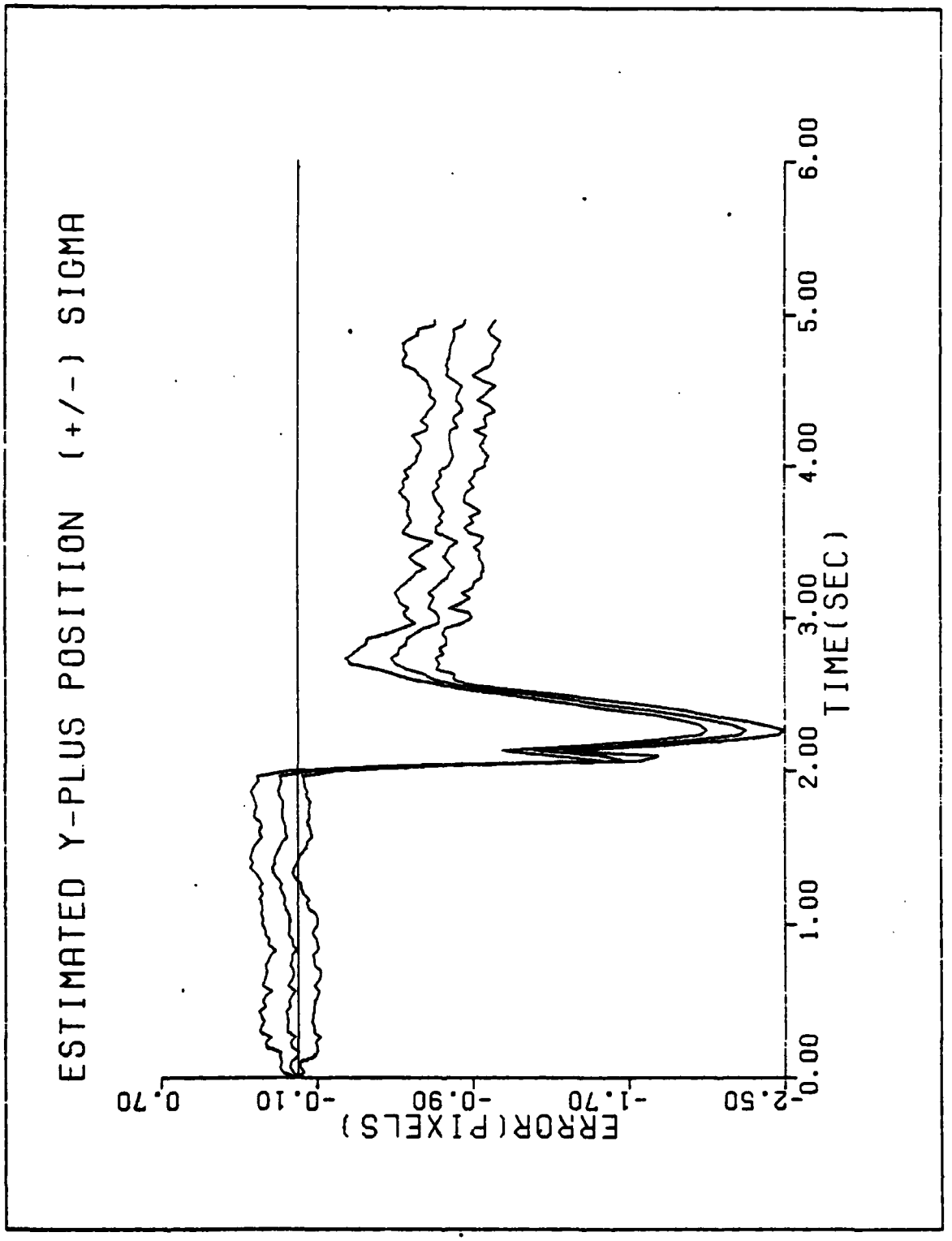


Figure B-9b. Performance Plot for G20T2MF

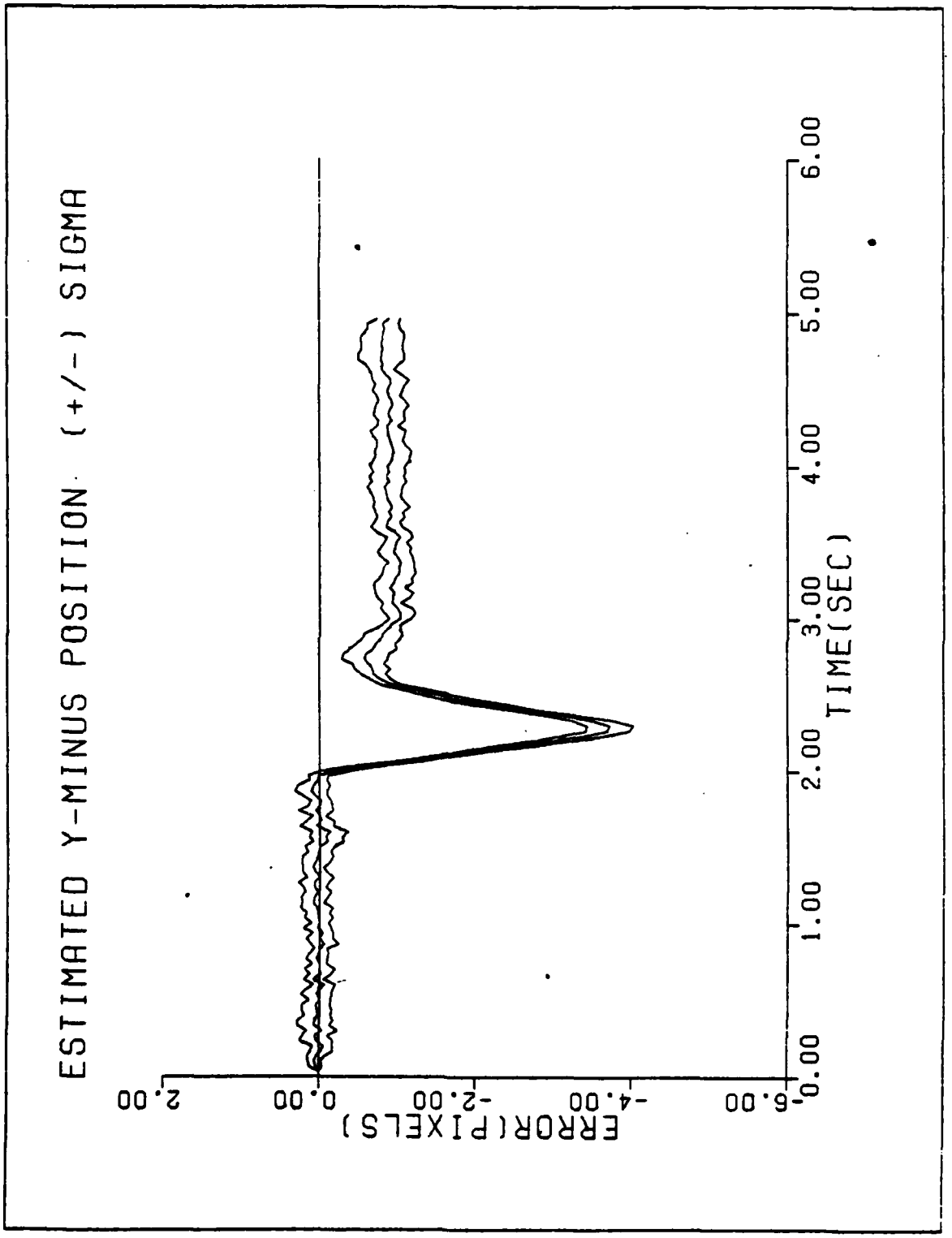


Figure B-10a. Performance Plot for G20T2F2

Figure B-10b. Performance Plot for G20T2F2

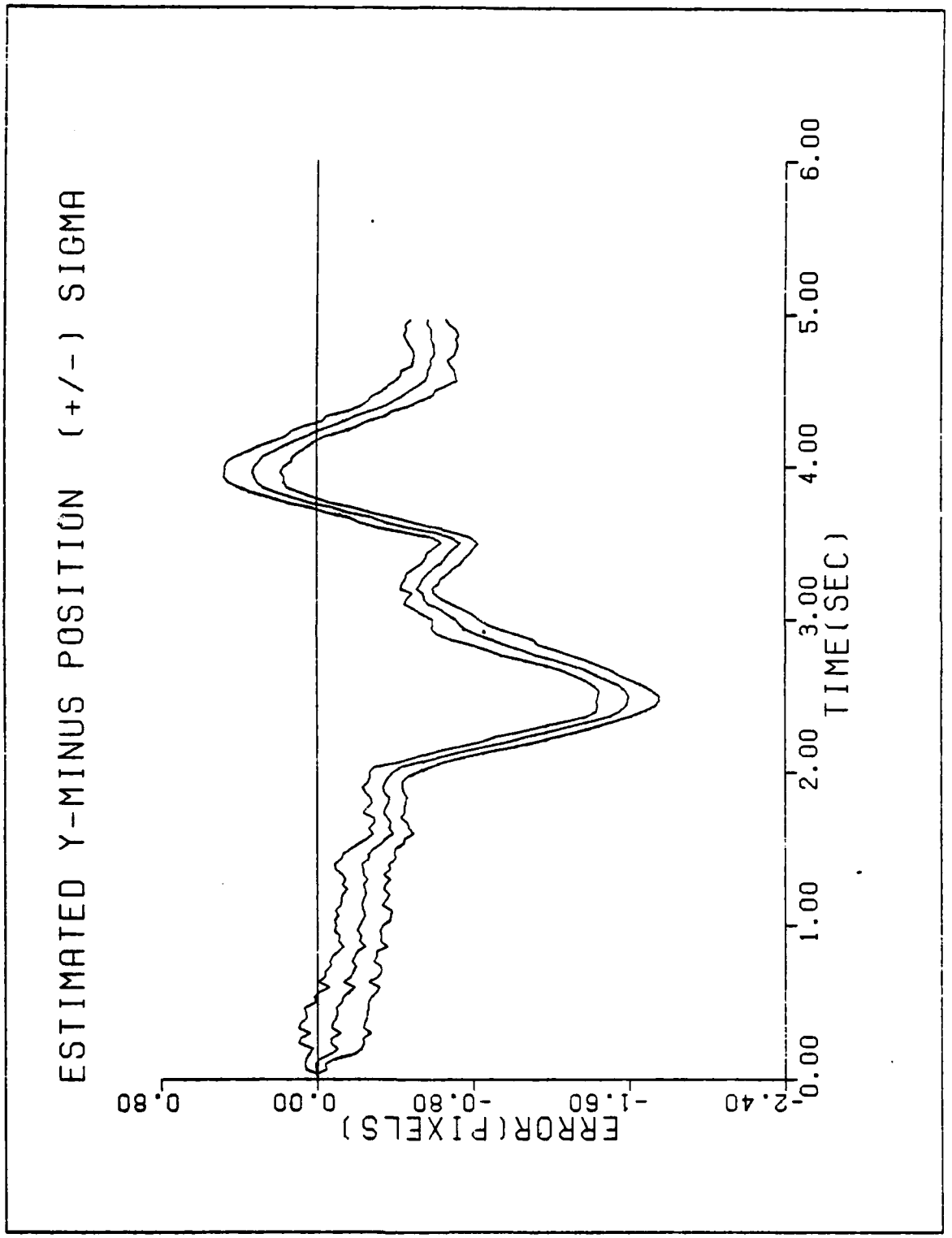


Figure B-11a. Performance Plot for G2T3MF

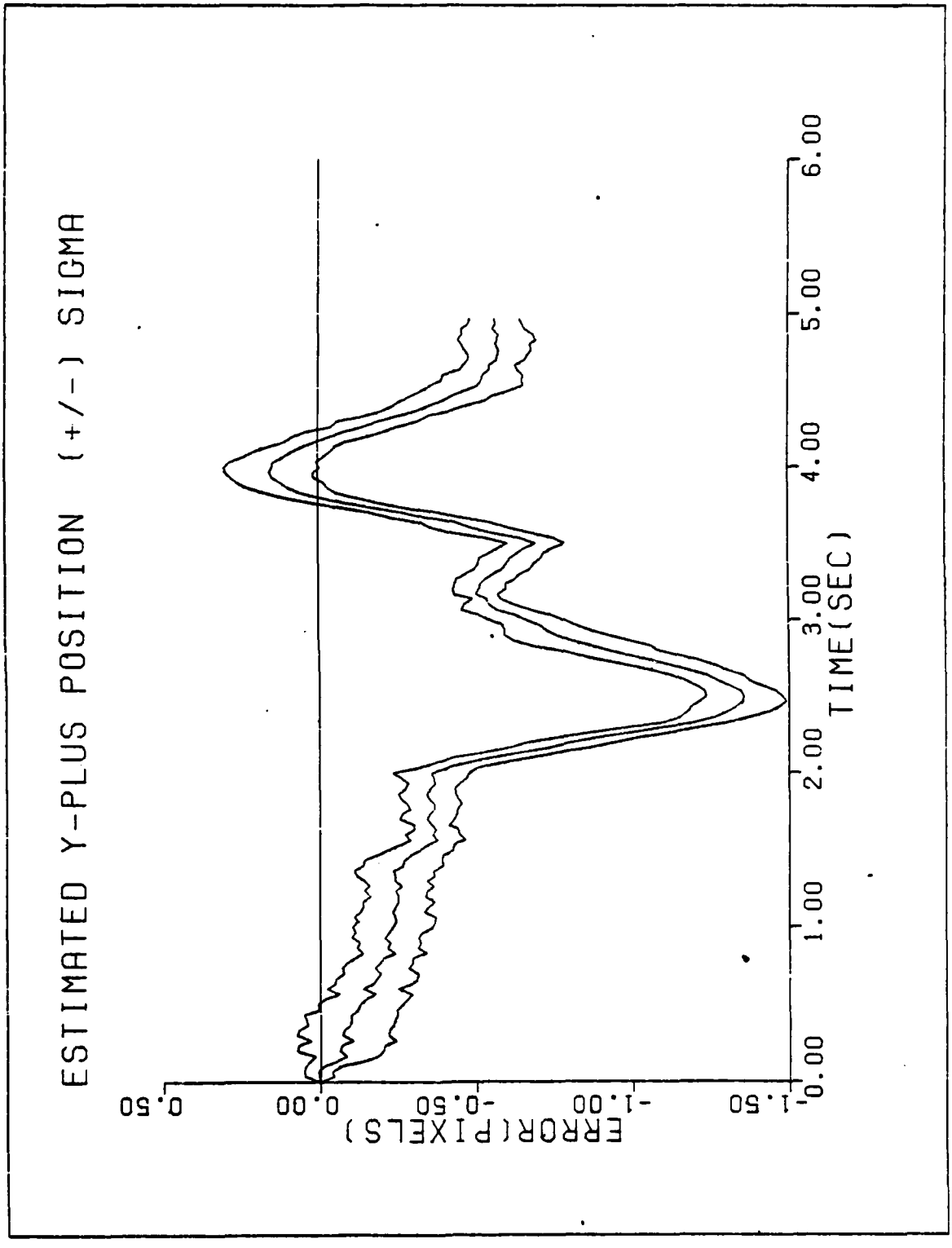


Figure B-11b. Performance Plot for G2T3MF

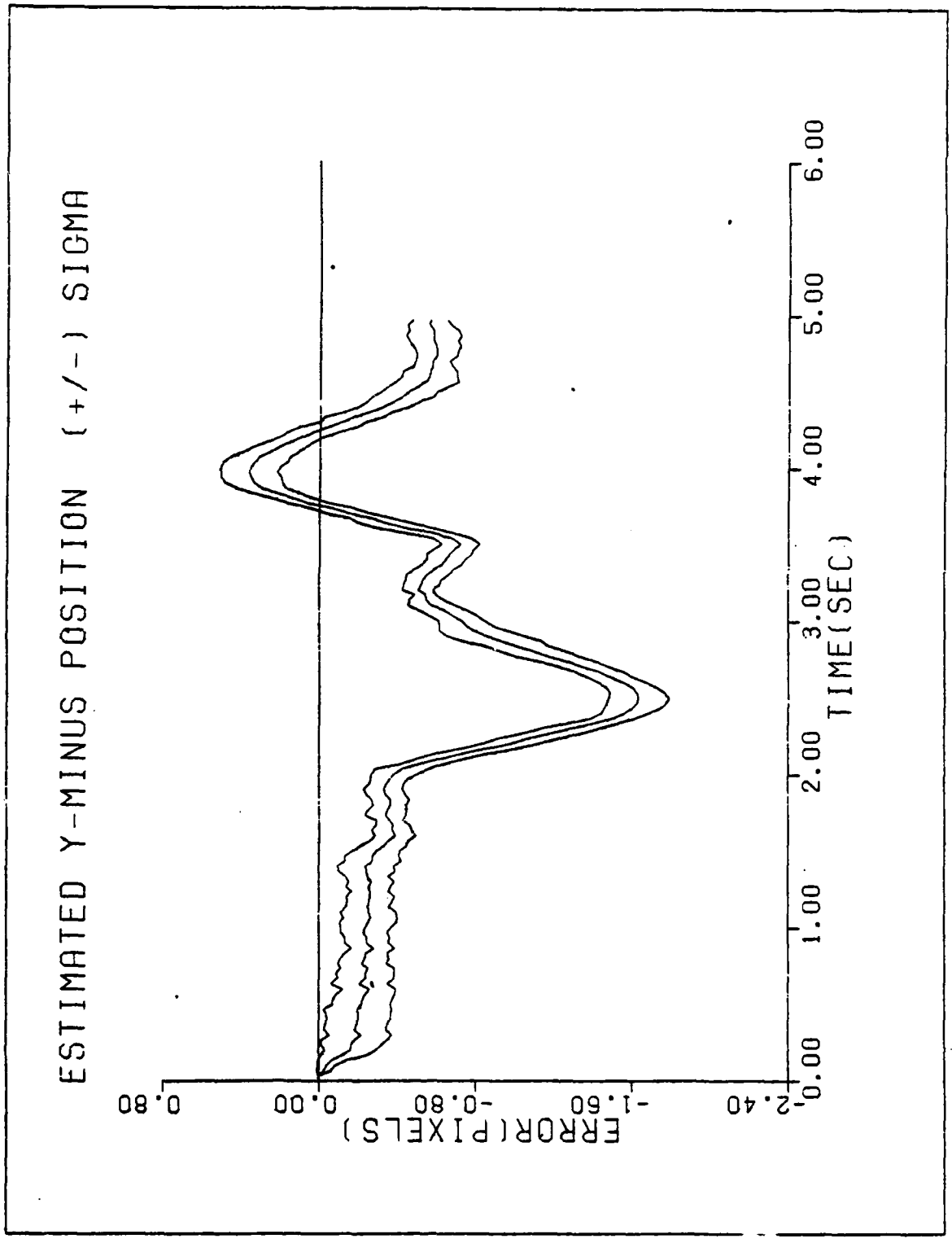


Figure B-12a. Performance Plot for G2T3F1

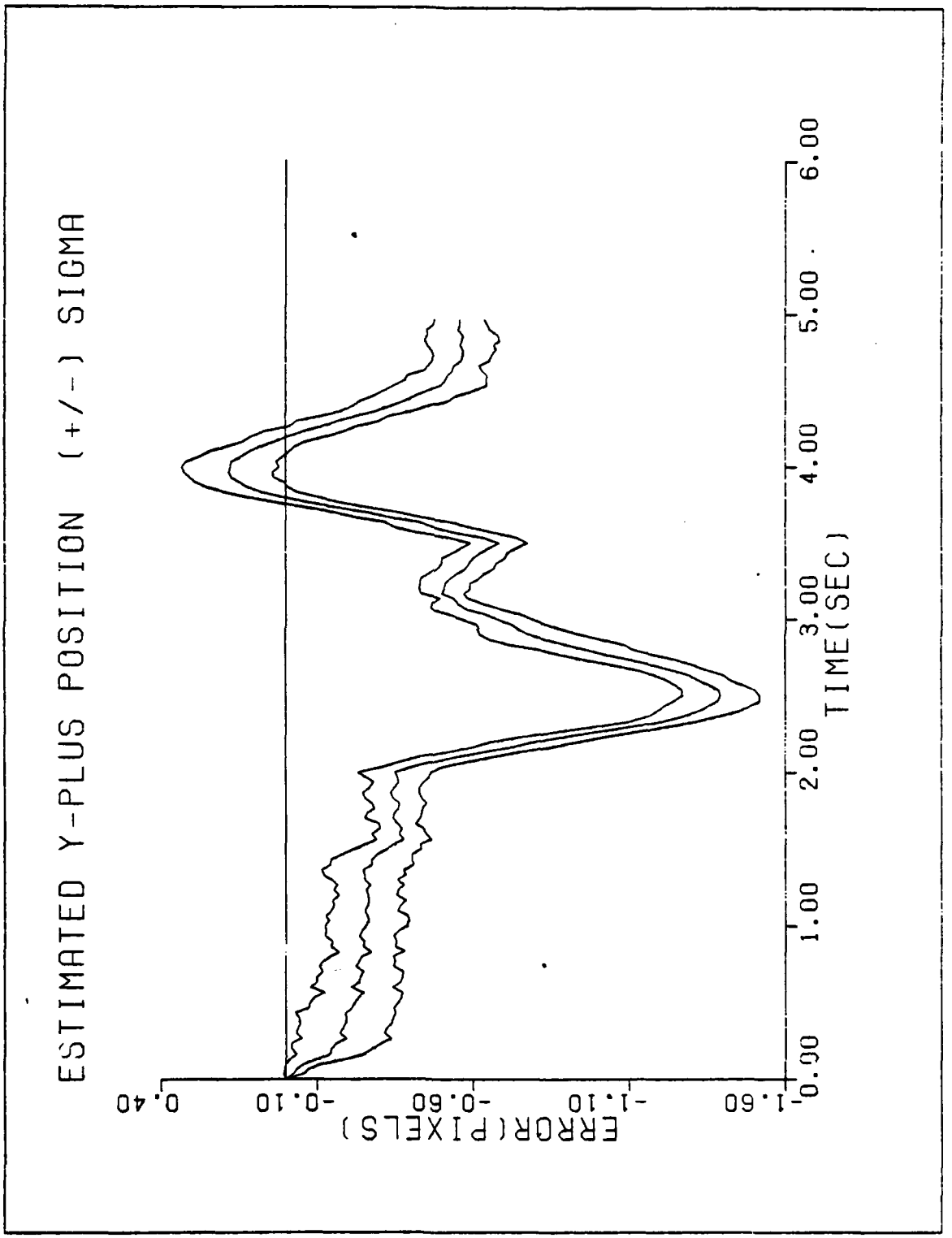


Figure B-12b. Performance Plot for G2T3F1

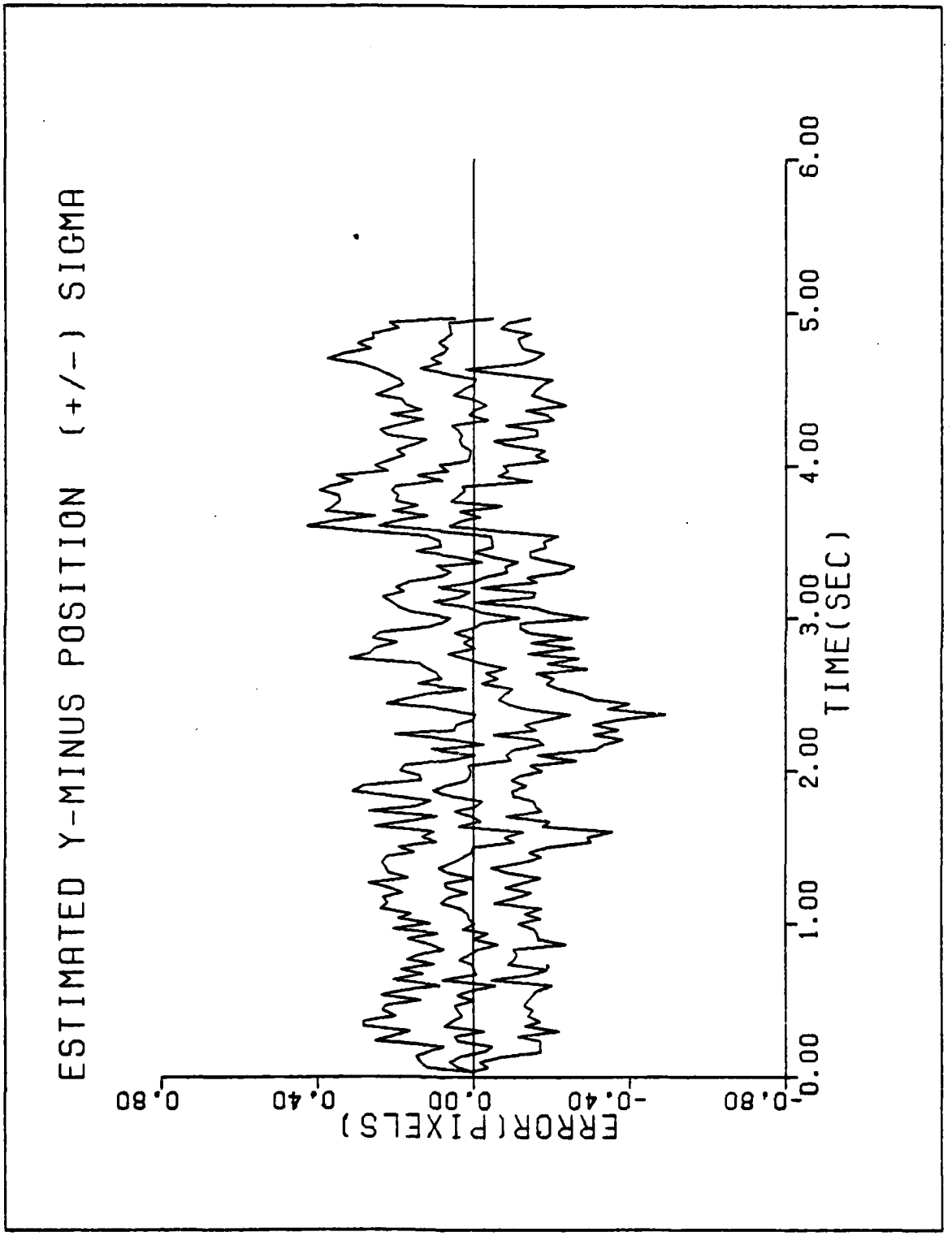


Figure B-13a. Performance Plot for G2T3F2

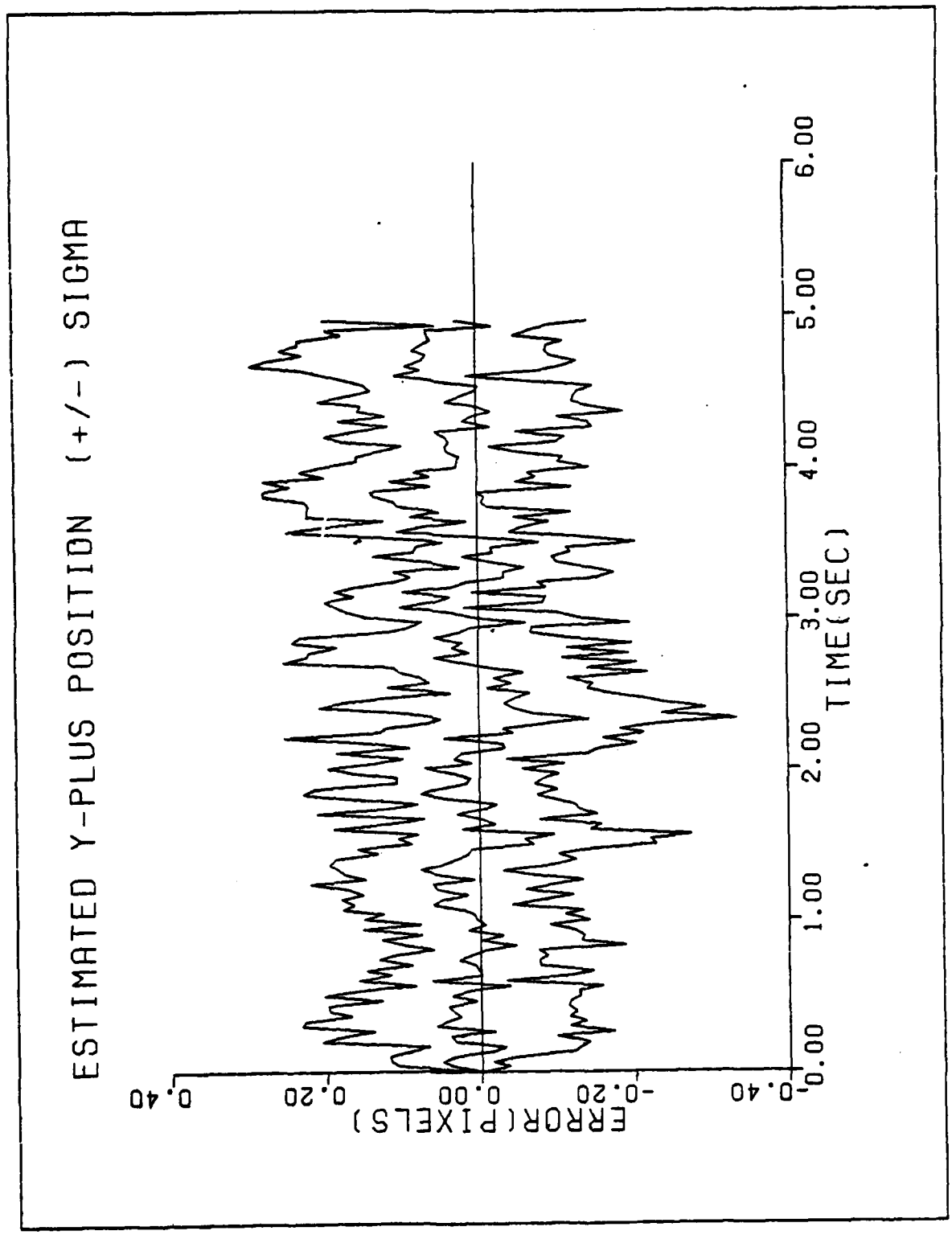


Figure B-13b. Performance Plot for G2T3F2

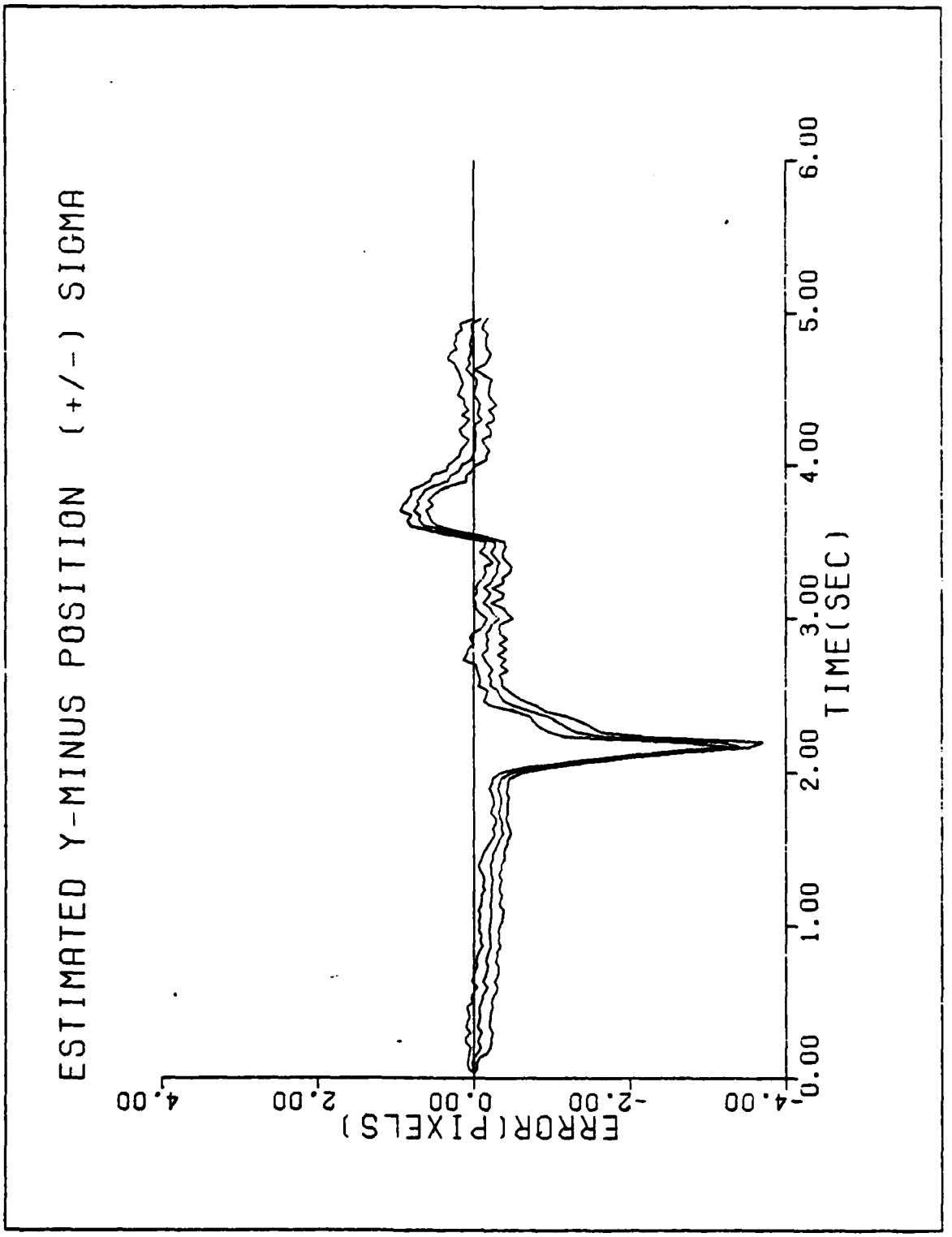


Figure B-14a. Performance Plot for G10T3MF

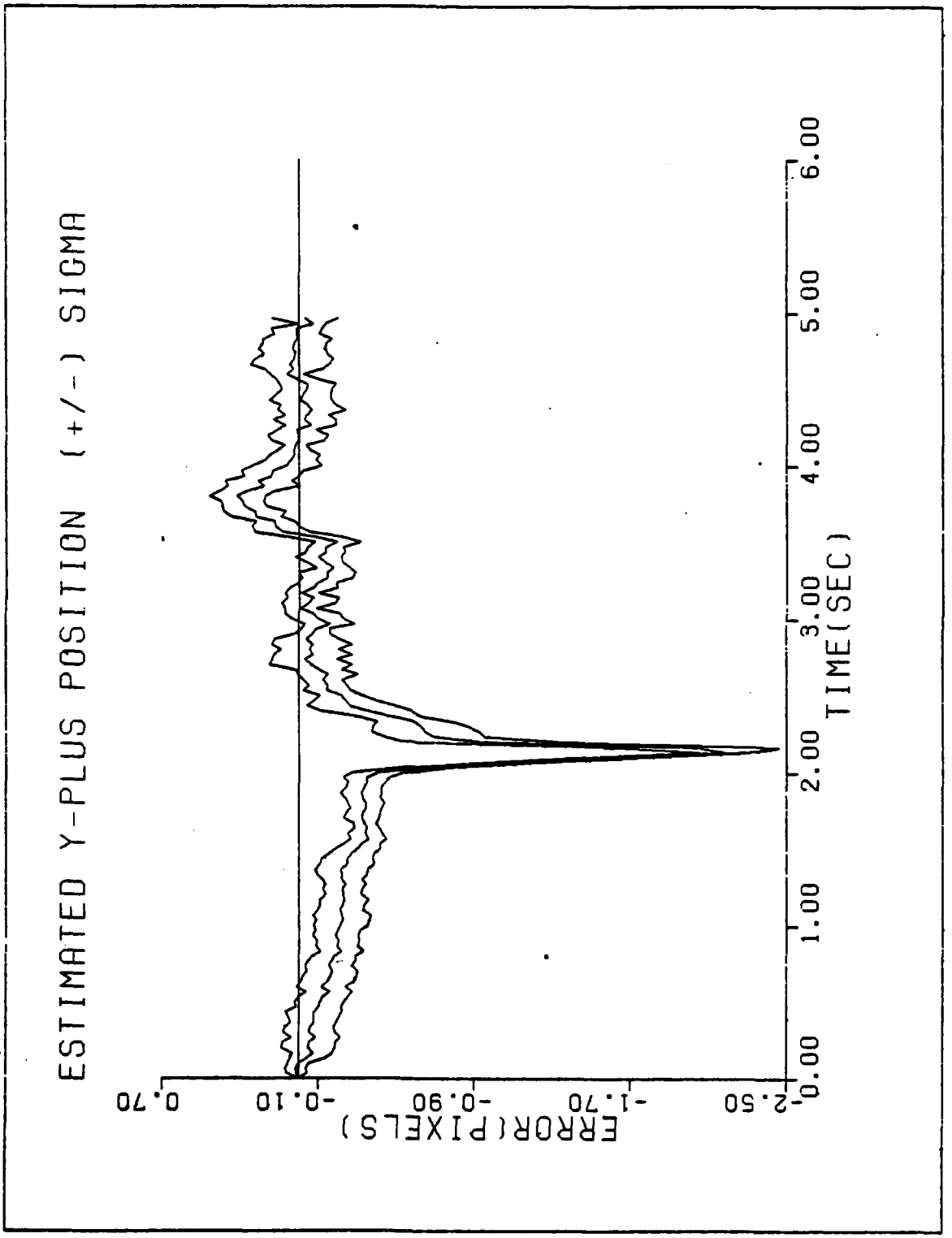


Figure B-14b. Performance Plot for G10T3MF

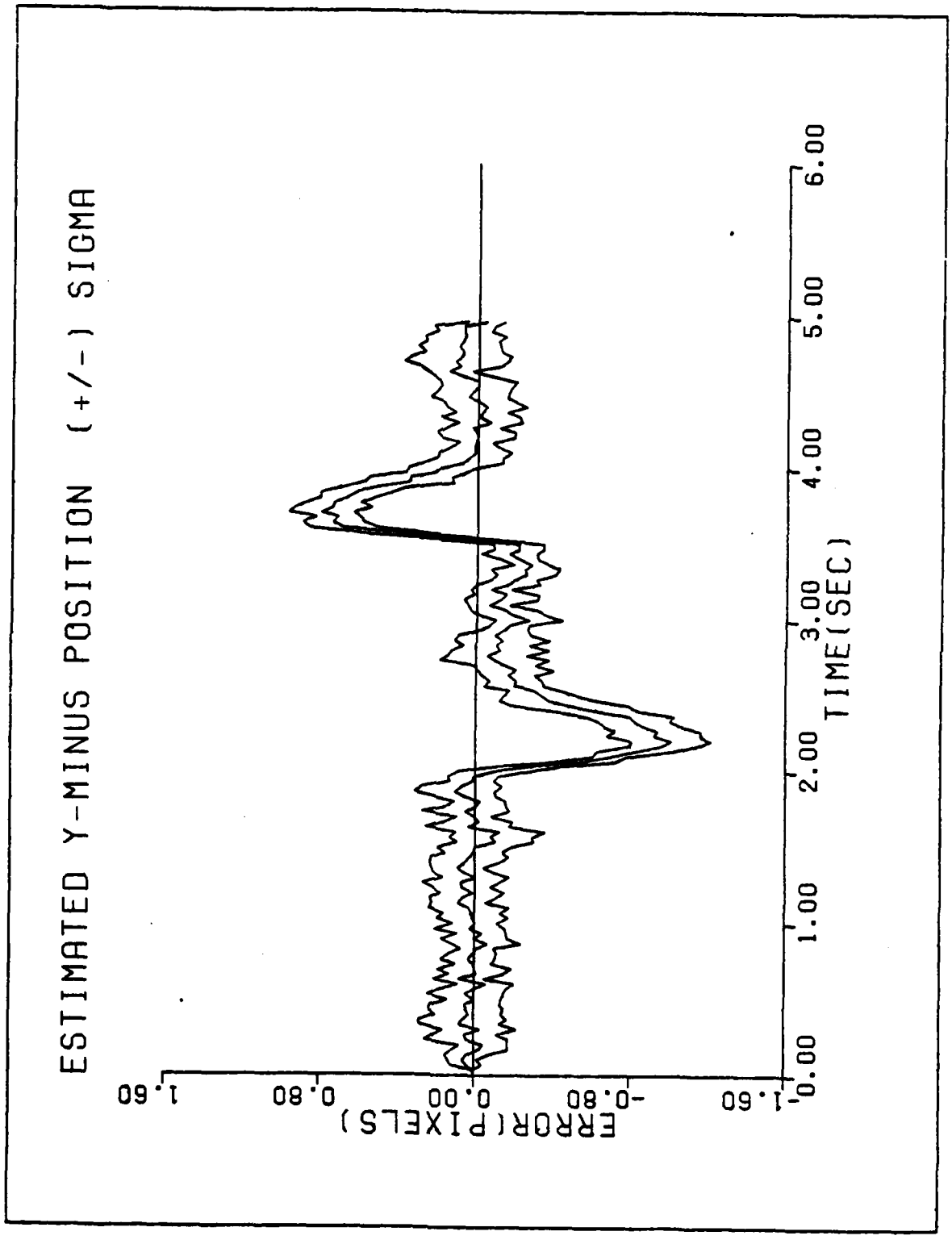


Figure B-15a. Performance Plot for G10T3F2

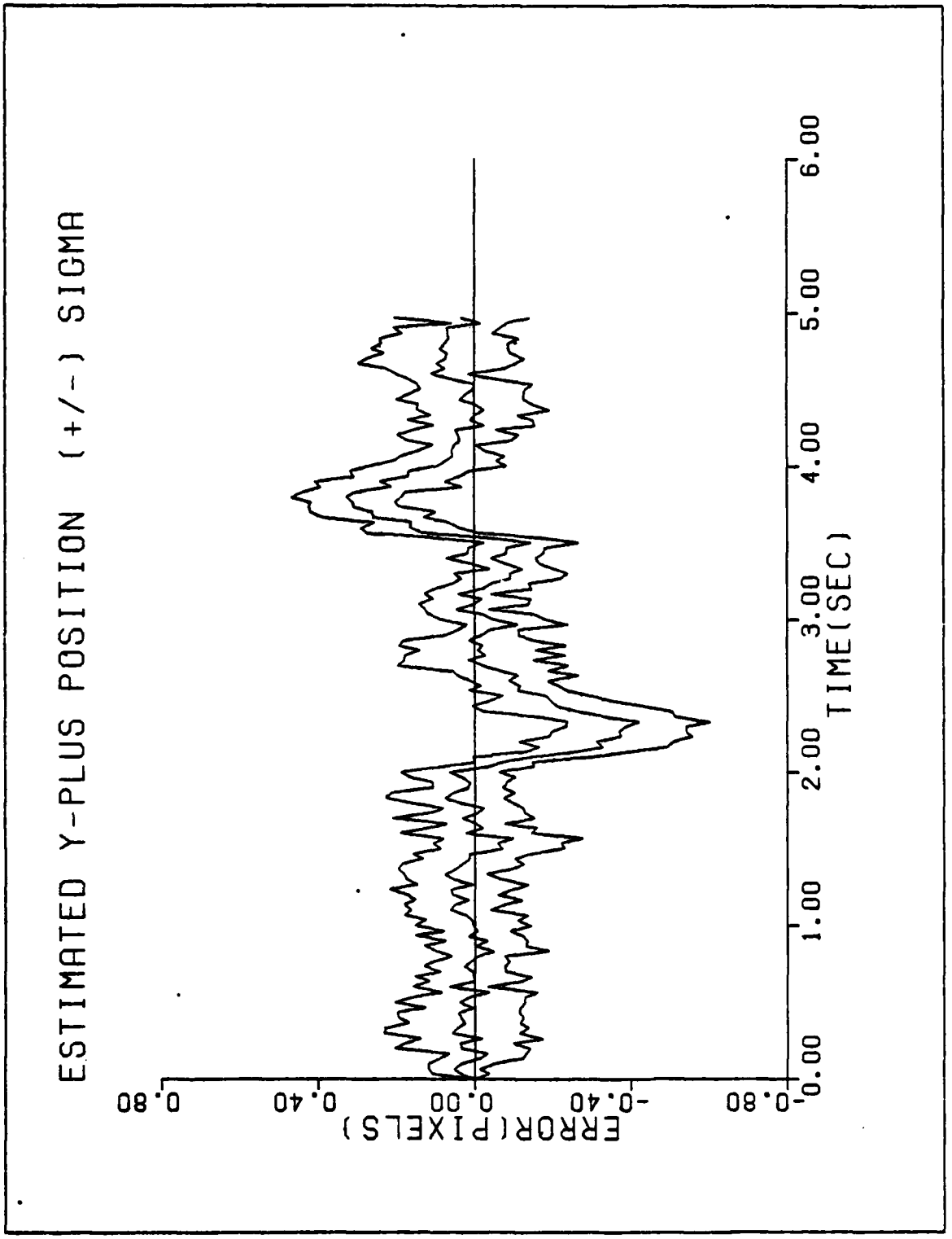


Figure B-15b. Performance Plot for G10T3F2

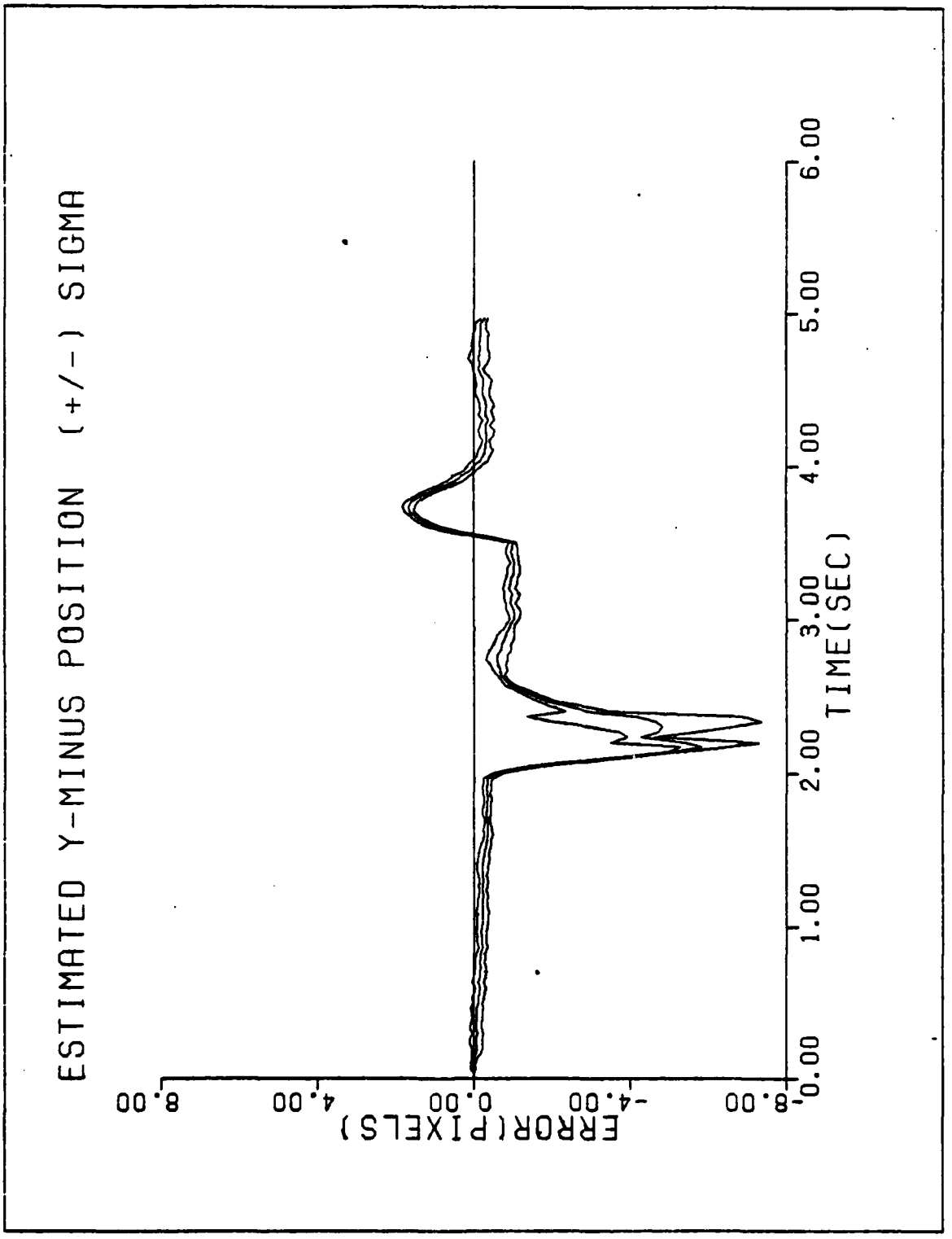


Figure B-16a. Performance Plot for G20T3MF

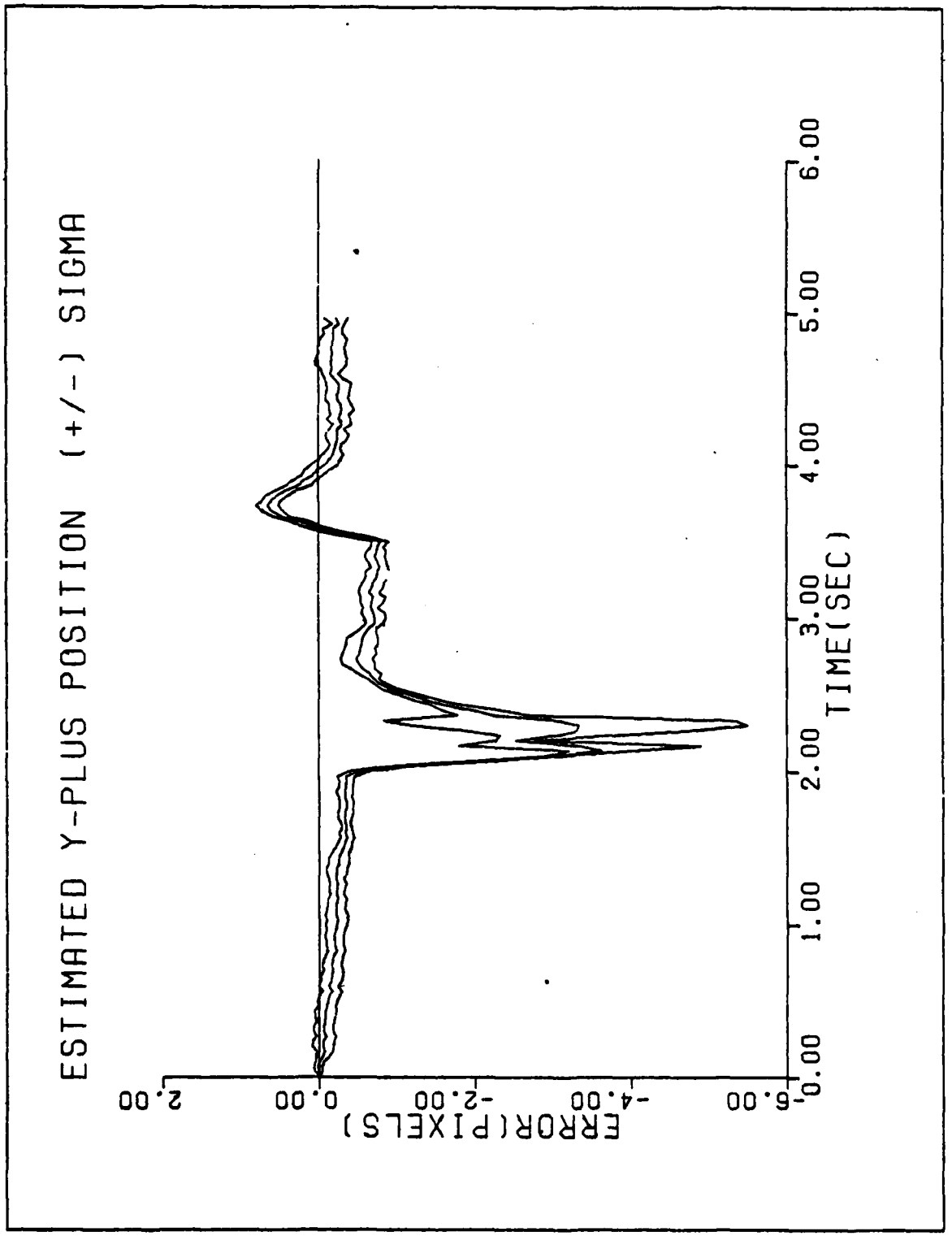


Figure B-16b. Performance Plot for G20T3MF

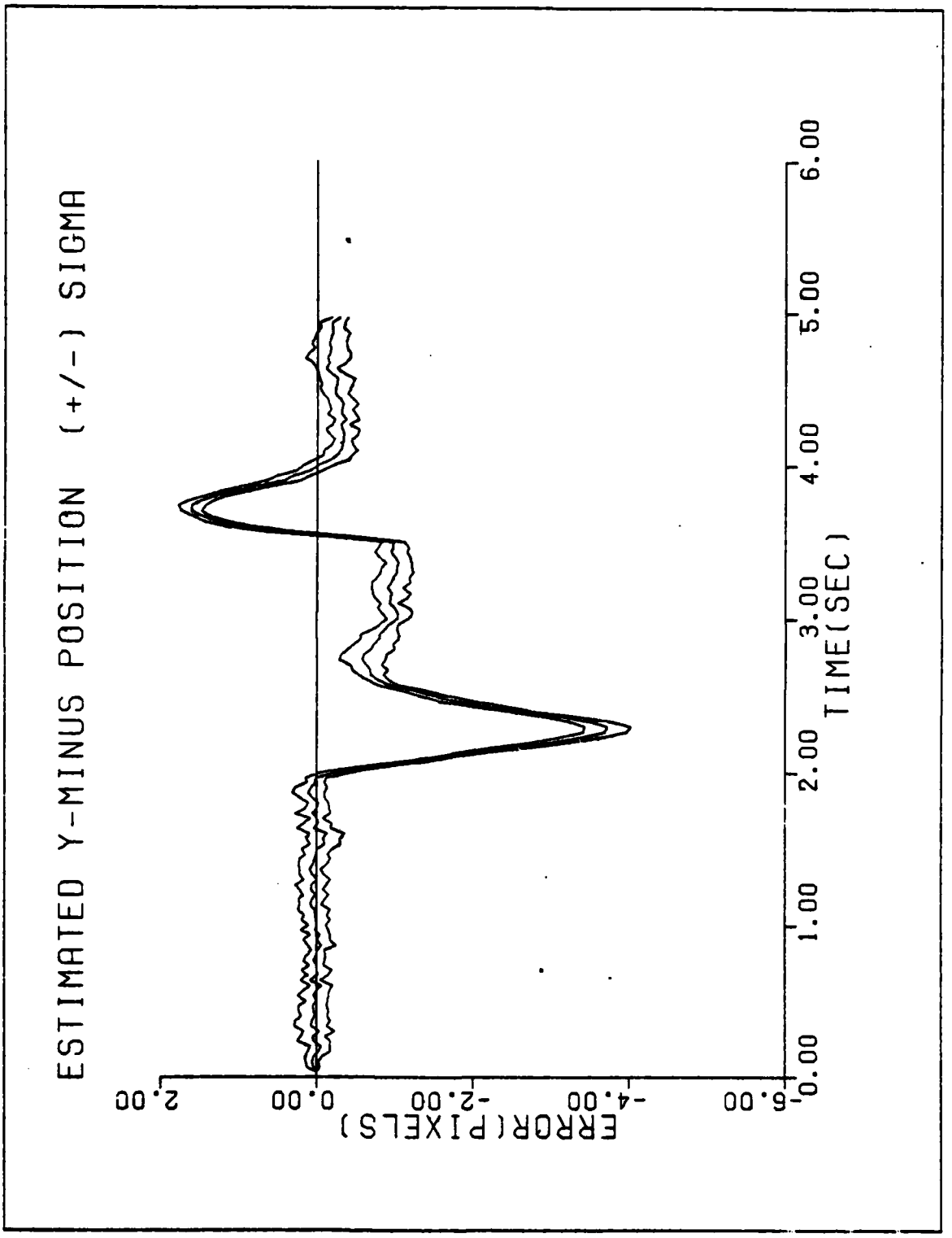


Figure B-17a. Performance Plot for G20T3F2

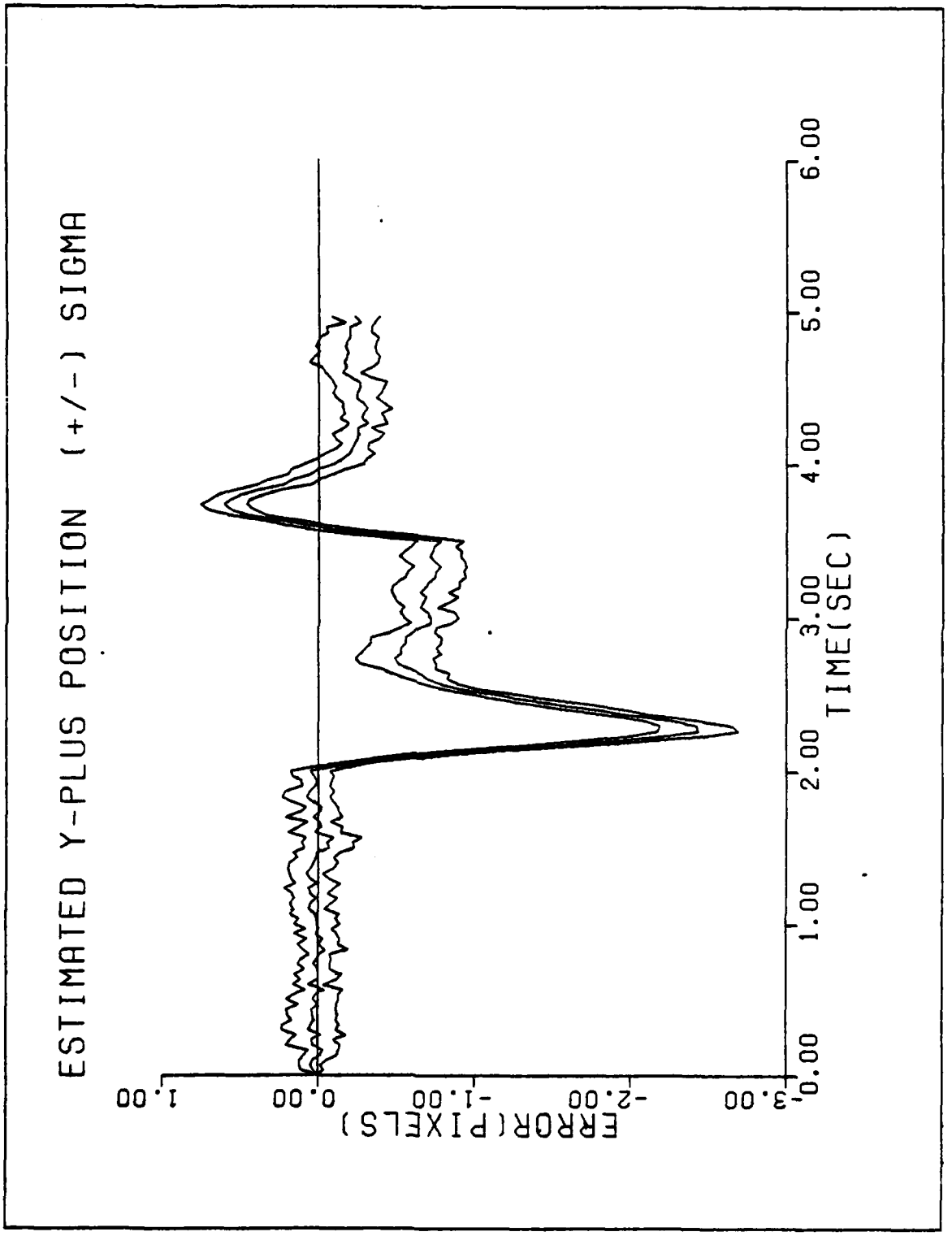


Figure B-17b. Performance Plot for G20T3F2

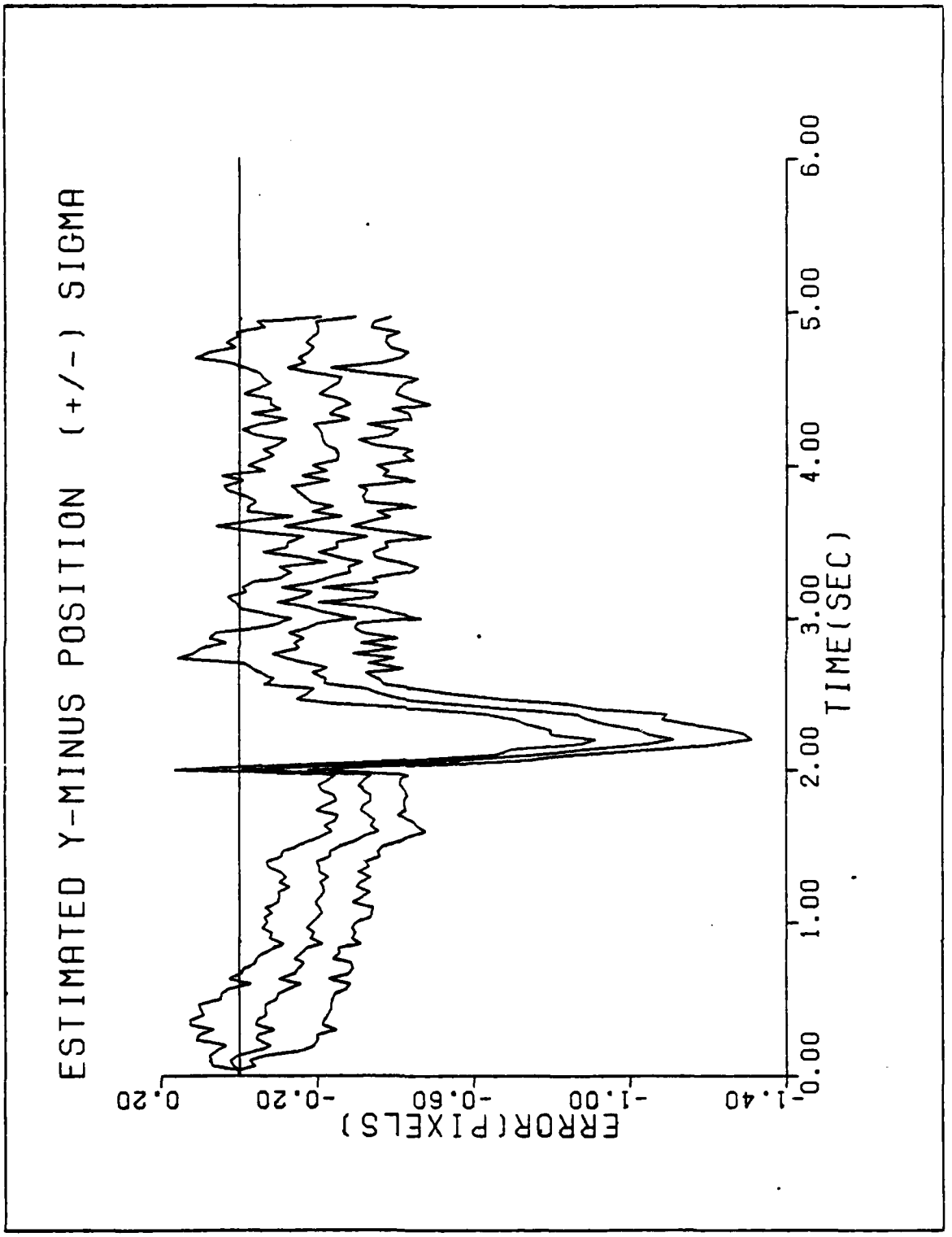


Figure B-18a. Performance Plot for GA10T2MF

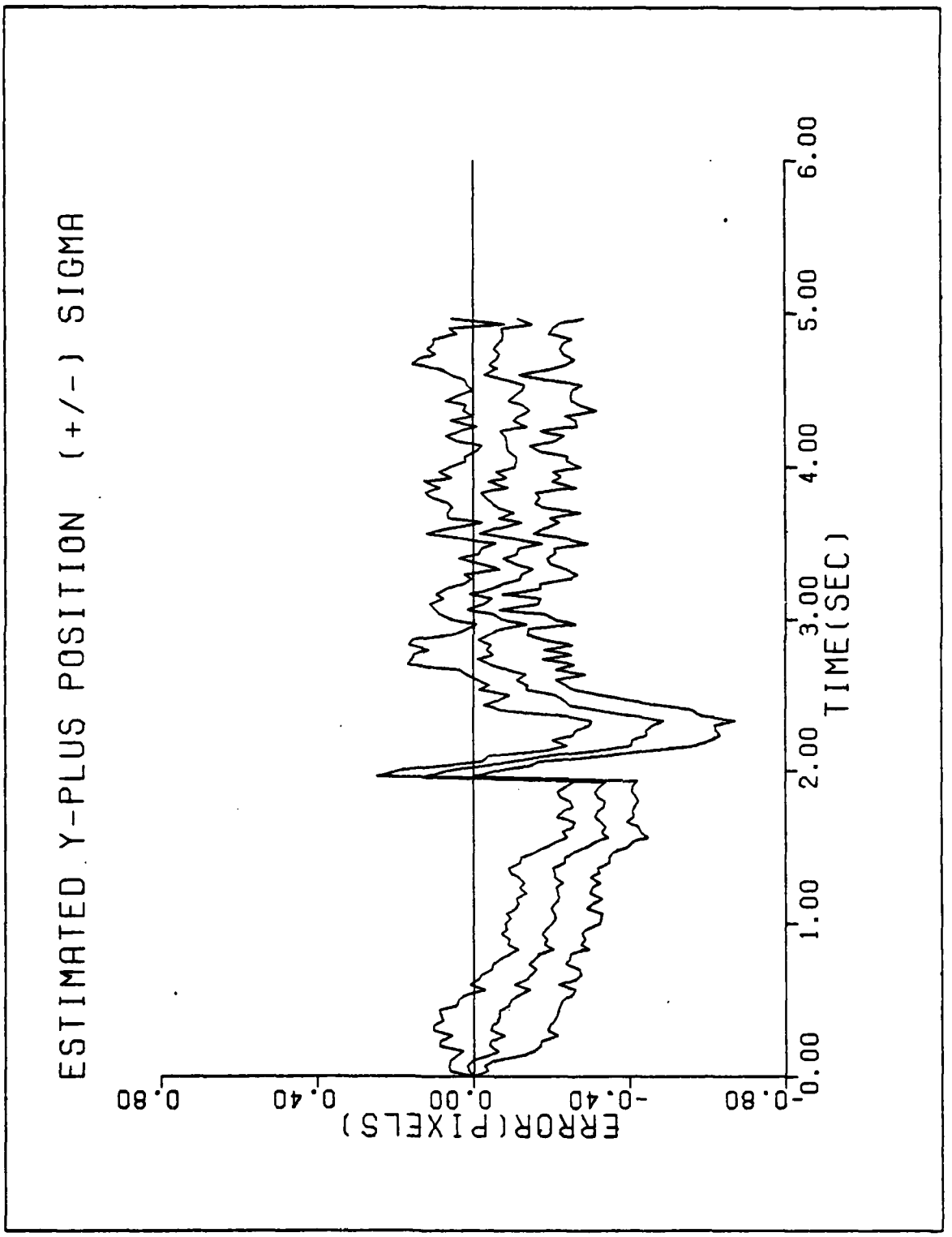


Figure B-18b. Performance Plot for GA10T2MF

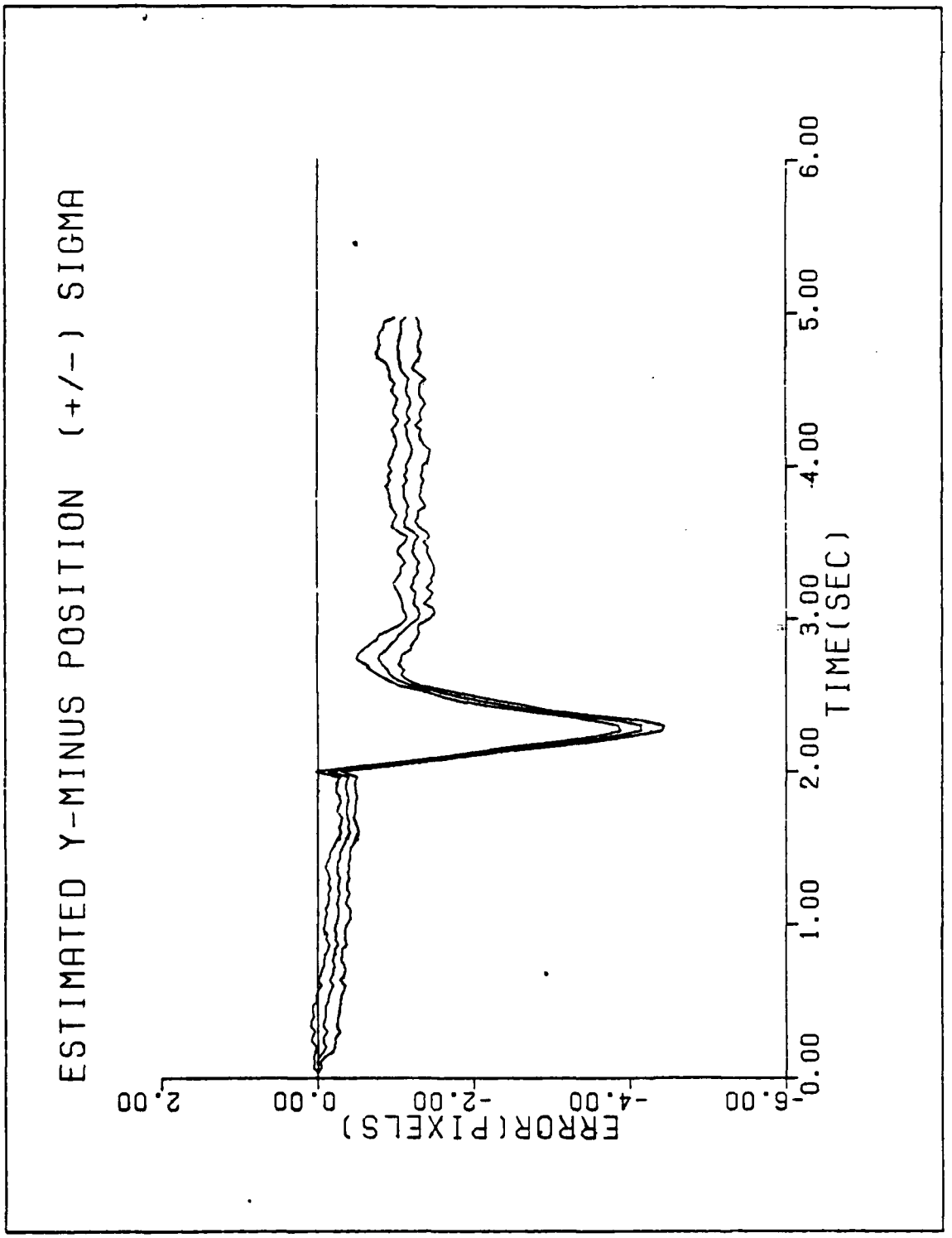


Figure B-19a. Performance Plot for GA20T2MF

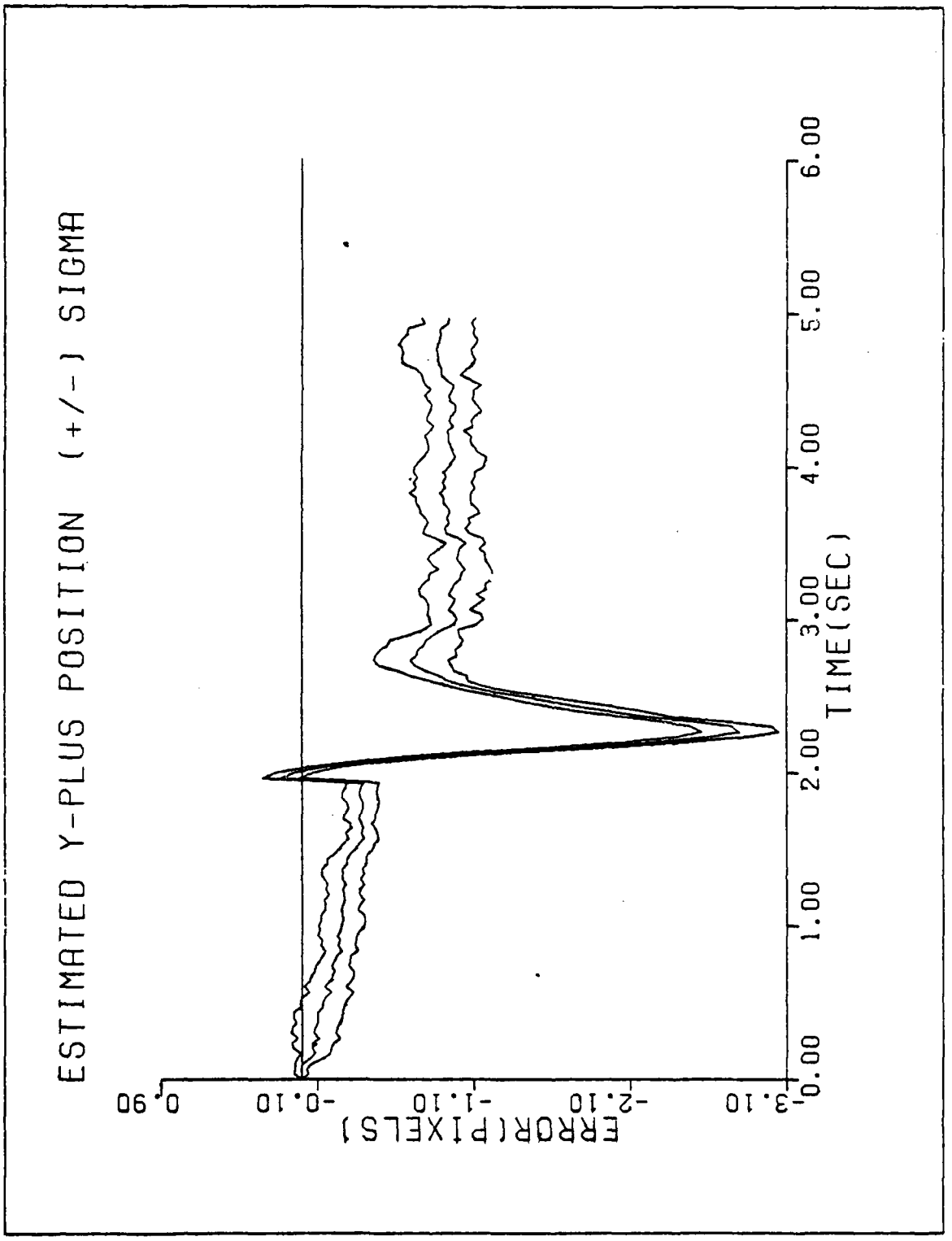


Figure B-19b. Performance Plot for GA20T2MF

Appendix C

This appendix contains the performance plots for the extended Kalman filter tracker configuration using a constant turn rate target acceleration model. Thes cases are identified according to the mnemonic code described in Figure V-3. The values for the parameters of these simulations are also found in Chapter V.

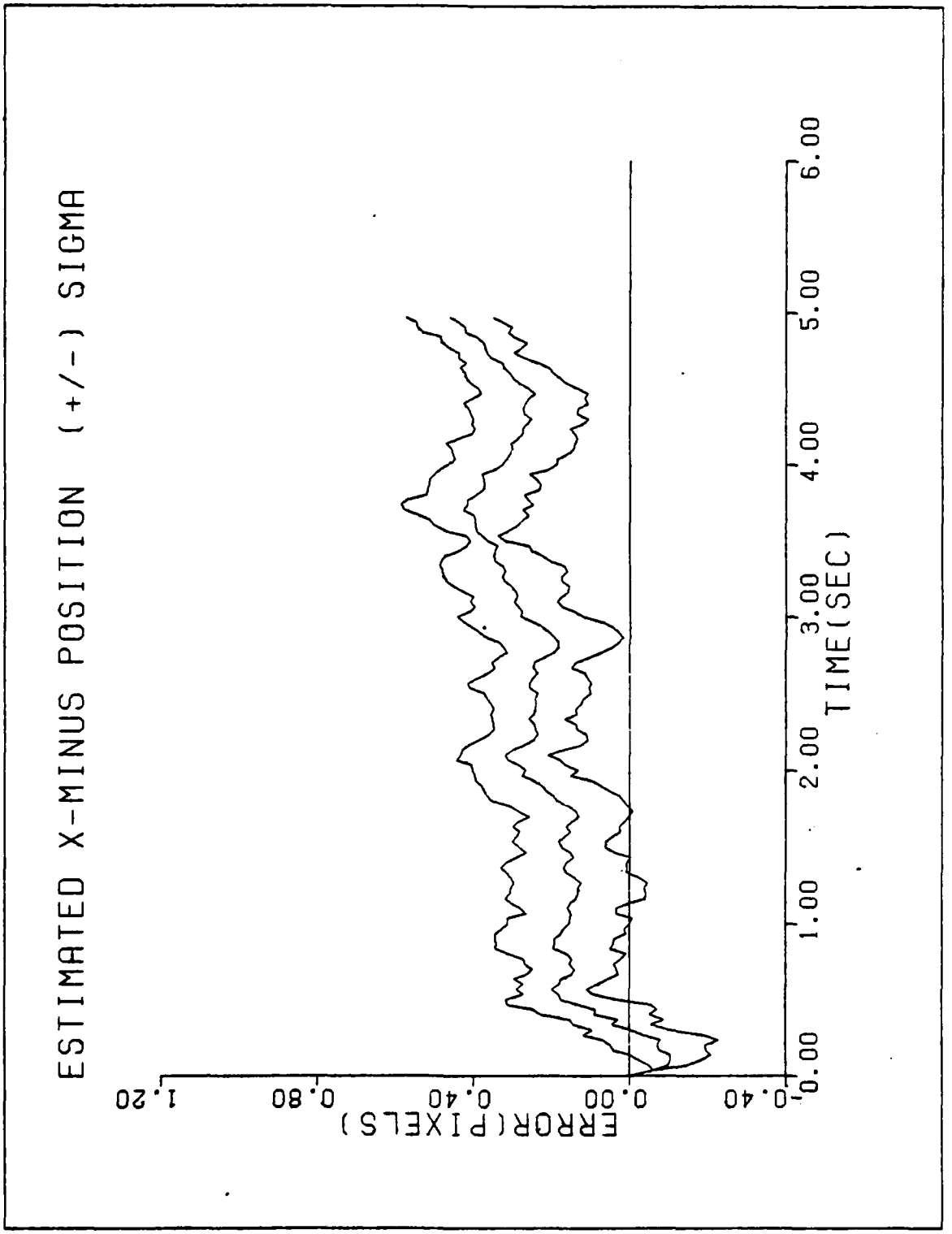


Figure C-1a. Performance Plot for CT1MF

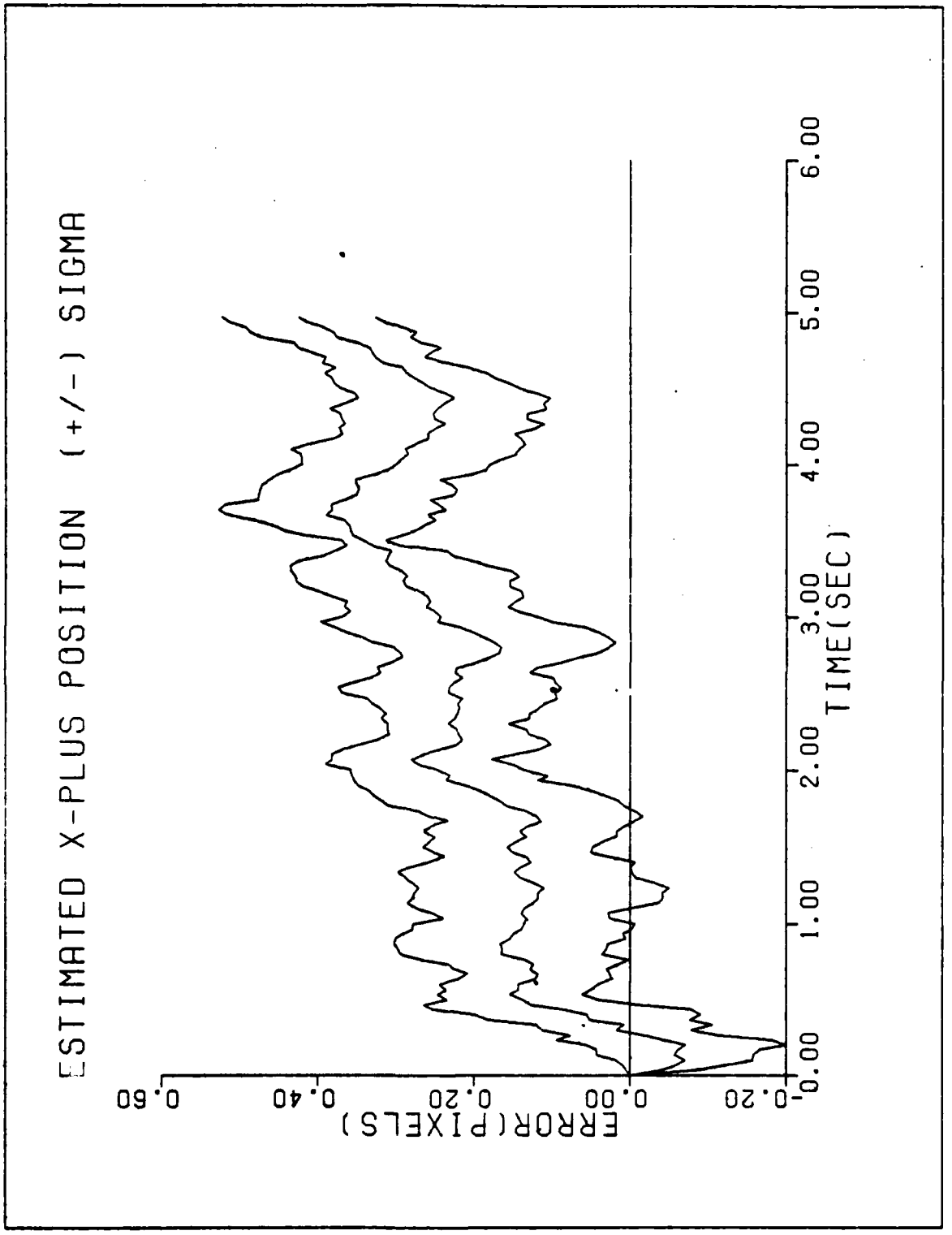


Figure C-1b. Performance Plot for CT1MF

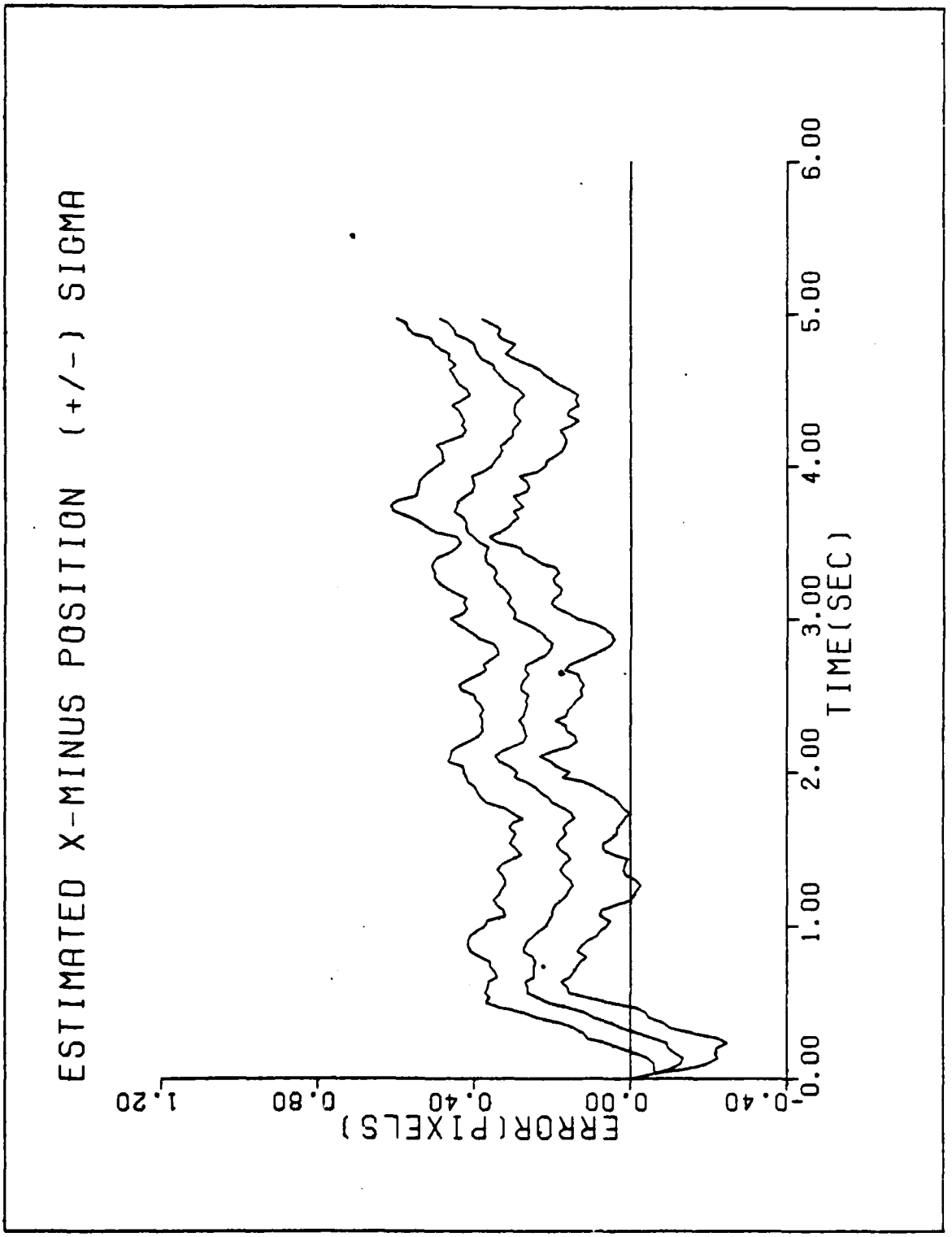


Figure C-2a. Performance Plot for CT1F1

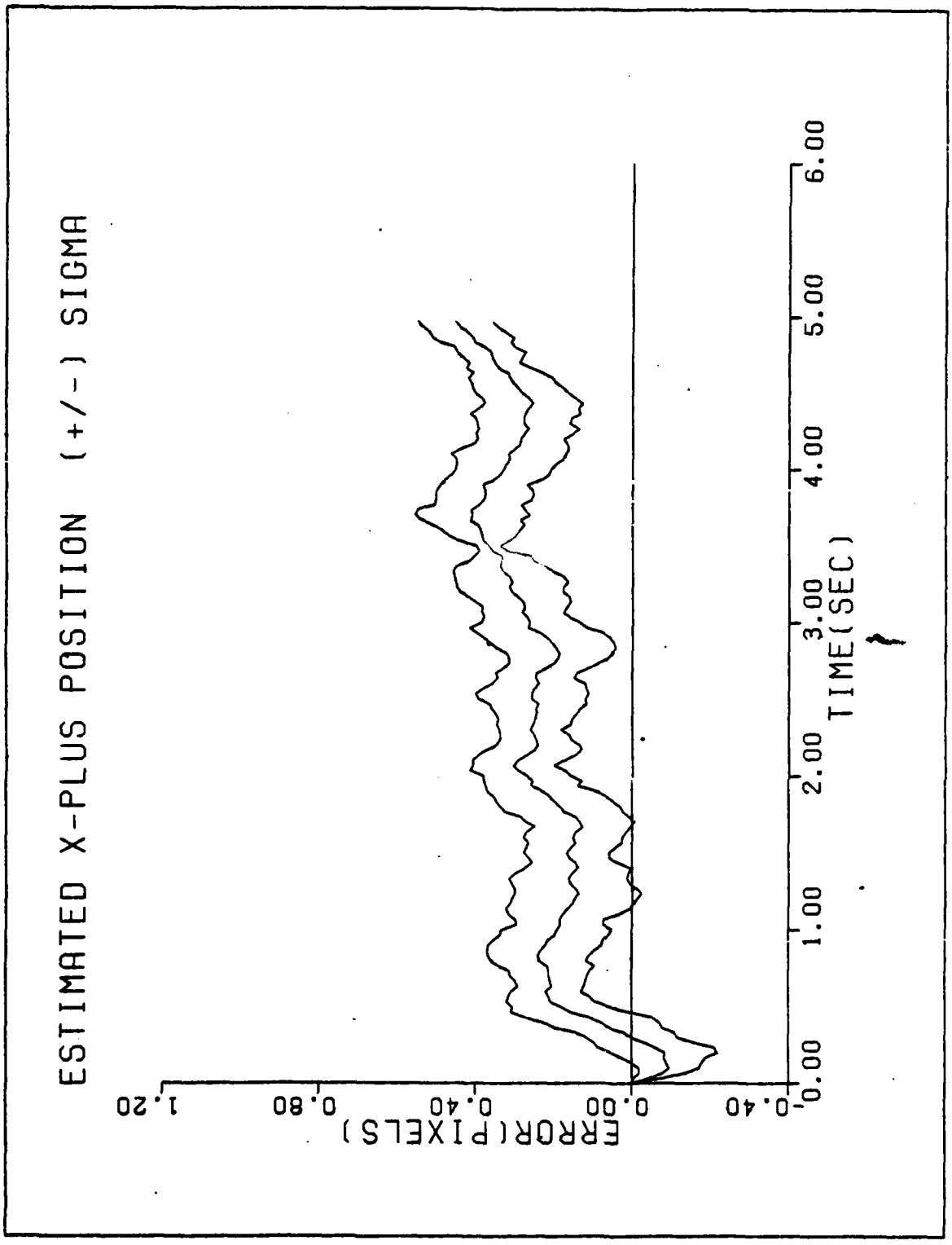


Figure C-2b. Performance Plot for CT1F1

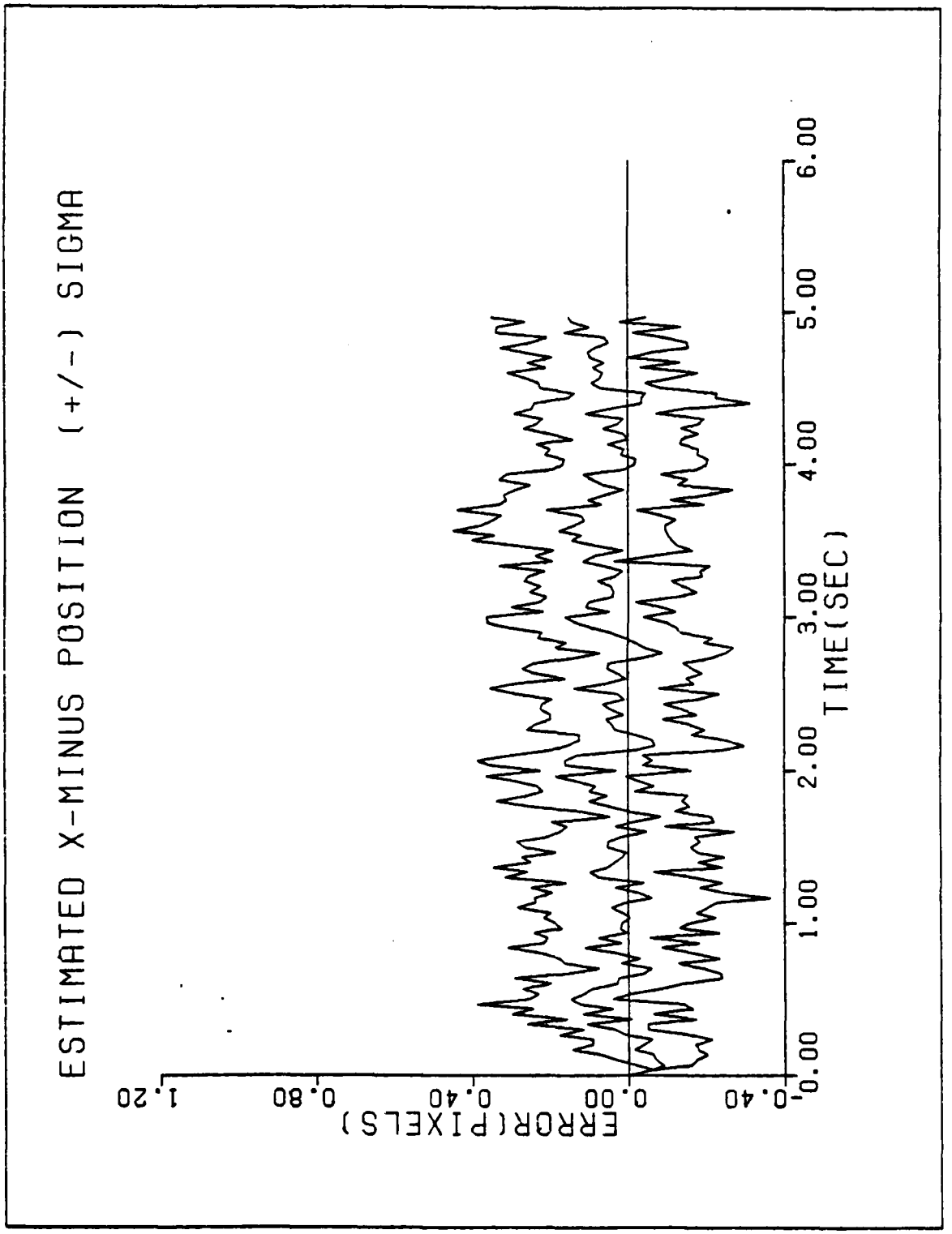


Figure C-3a. Performance Plot for CT1F2

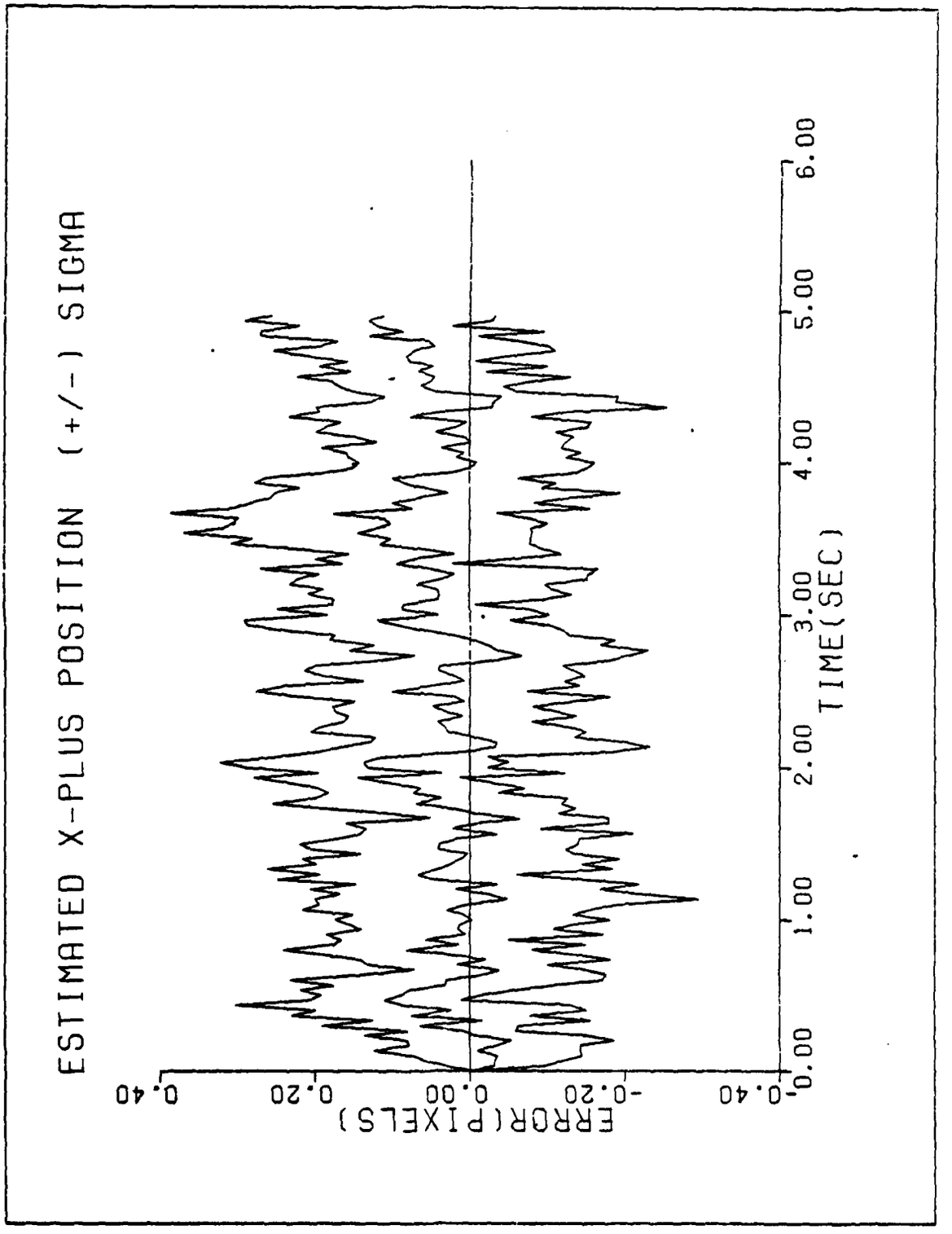


Figure C-3b. Performance Plot for CT1F2

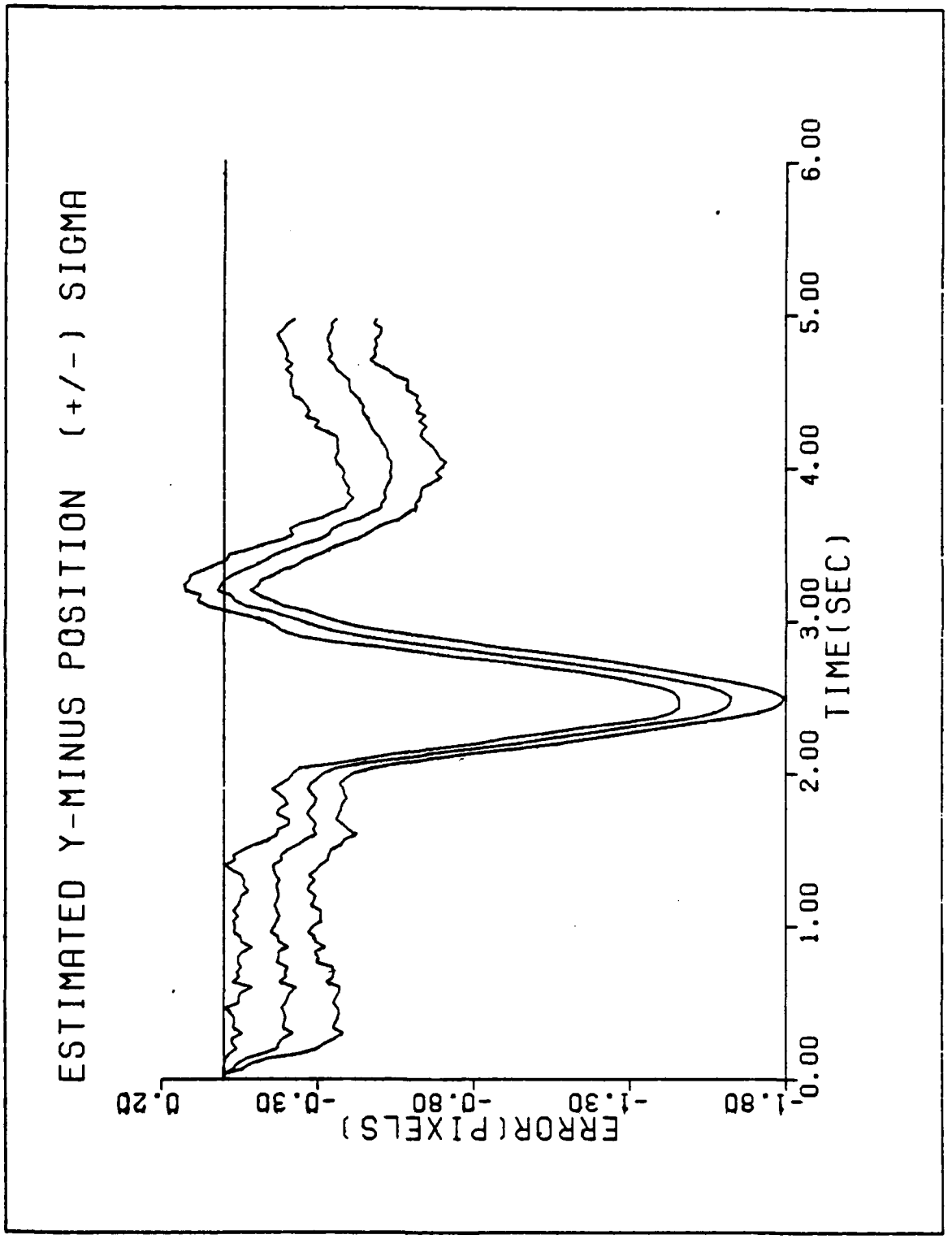


Figure C-4a. Performance Plot for C2T2MF

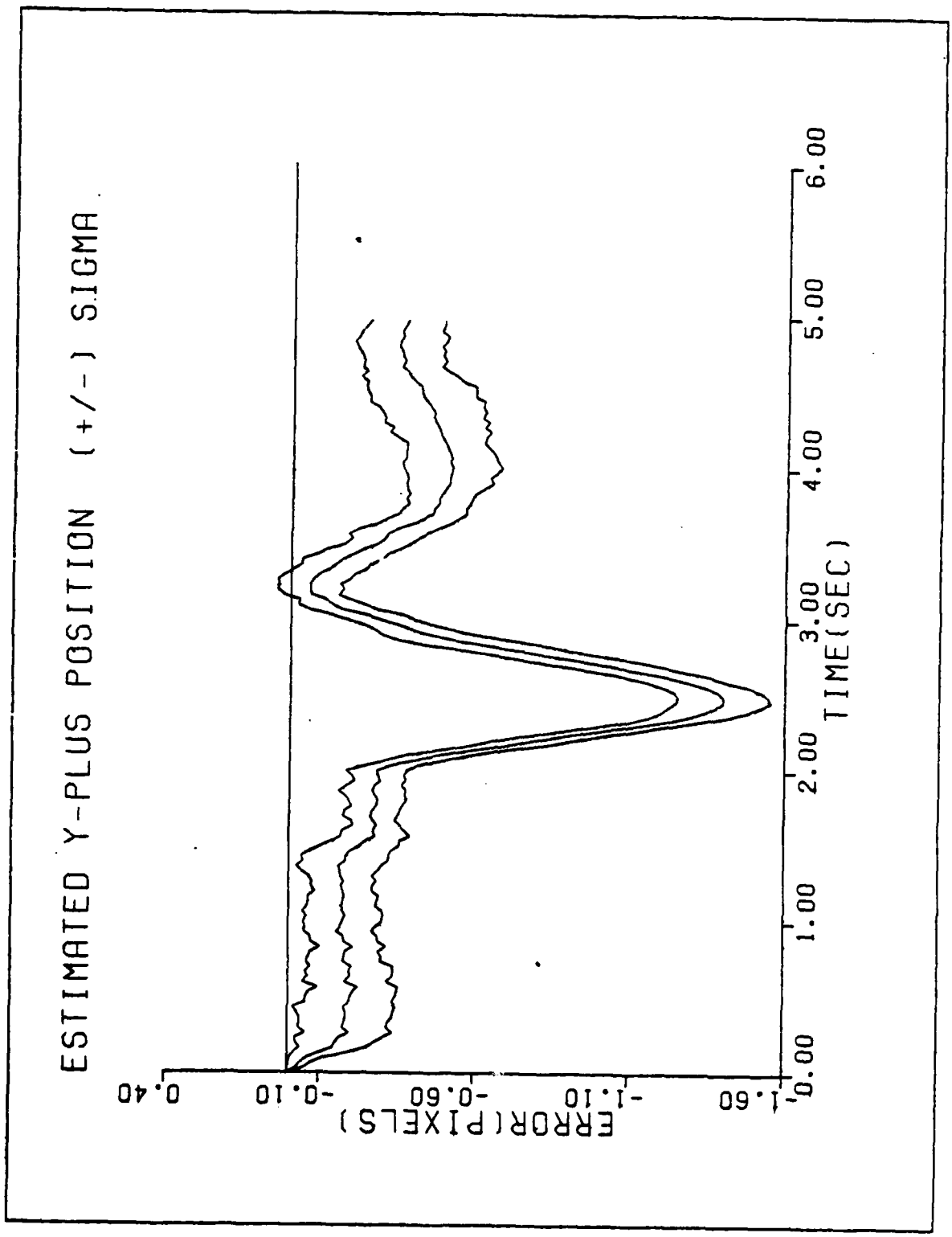


Figure C-4b. Performance Plot for C2T2MF

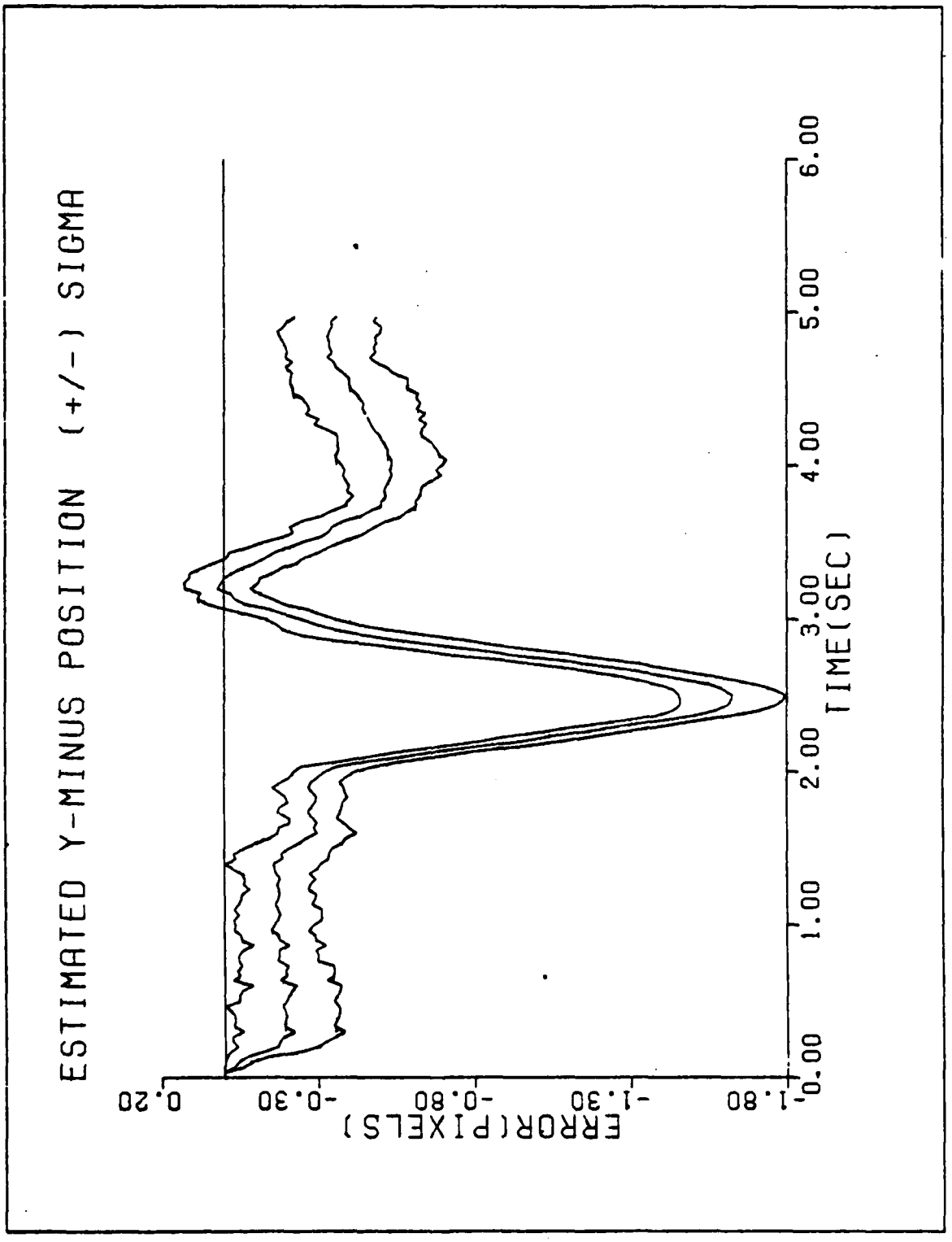


Figure C-5a. Performance Plot for C2T2F1

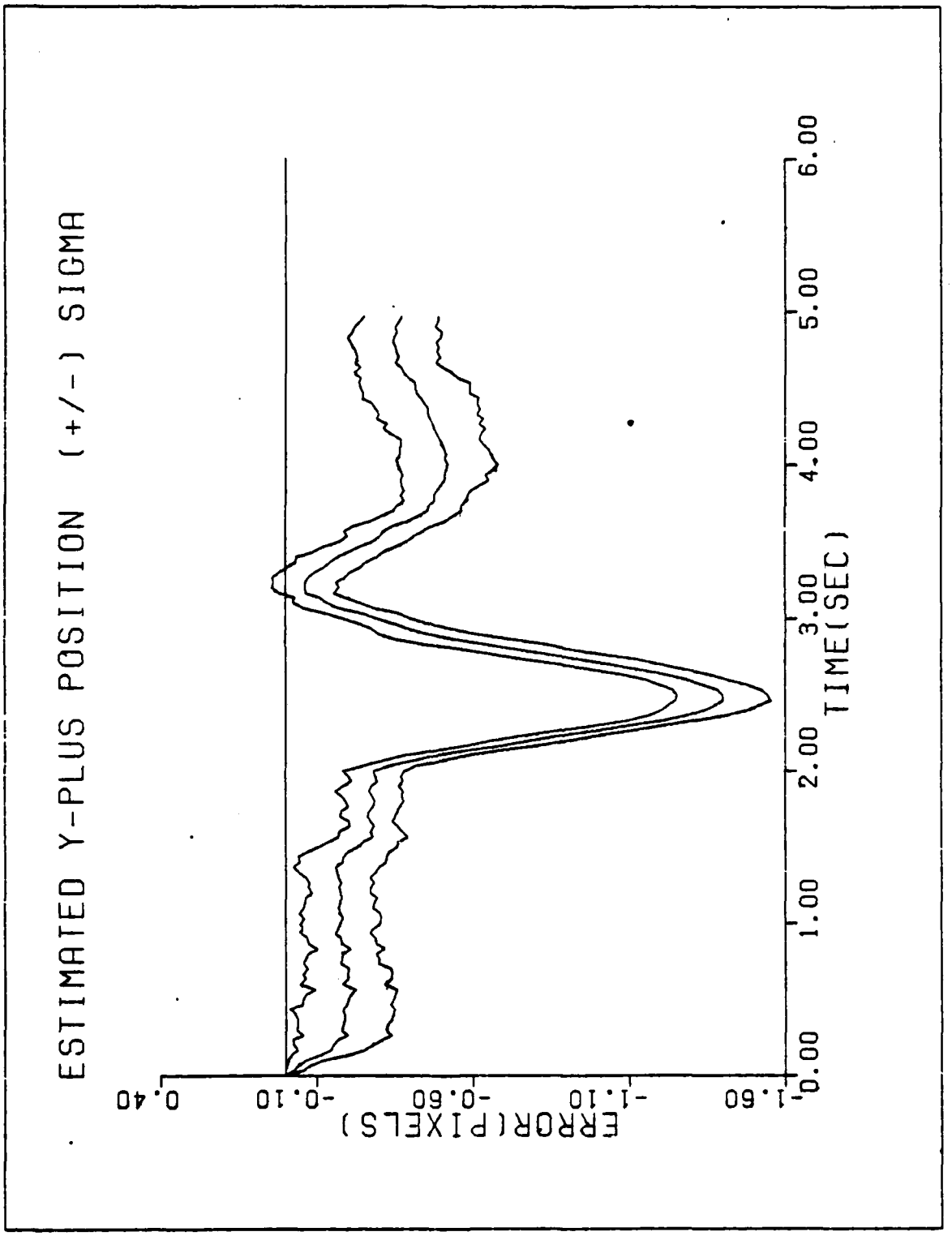


Figure C-5b. Performance Plot for C2T2F1

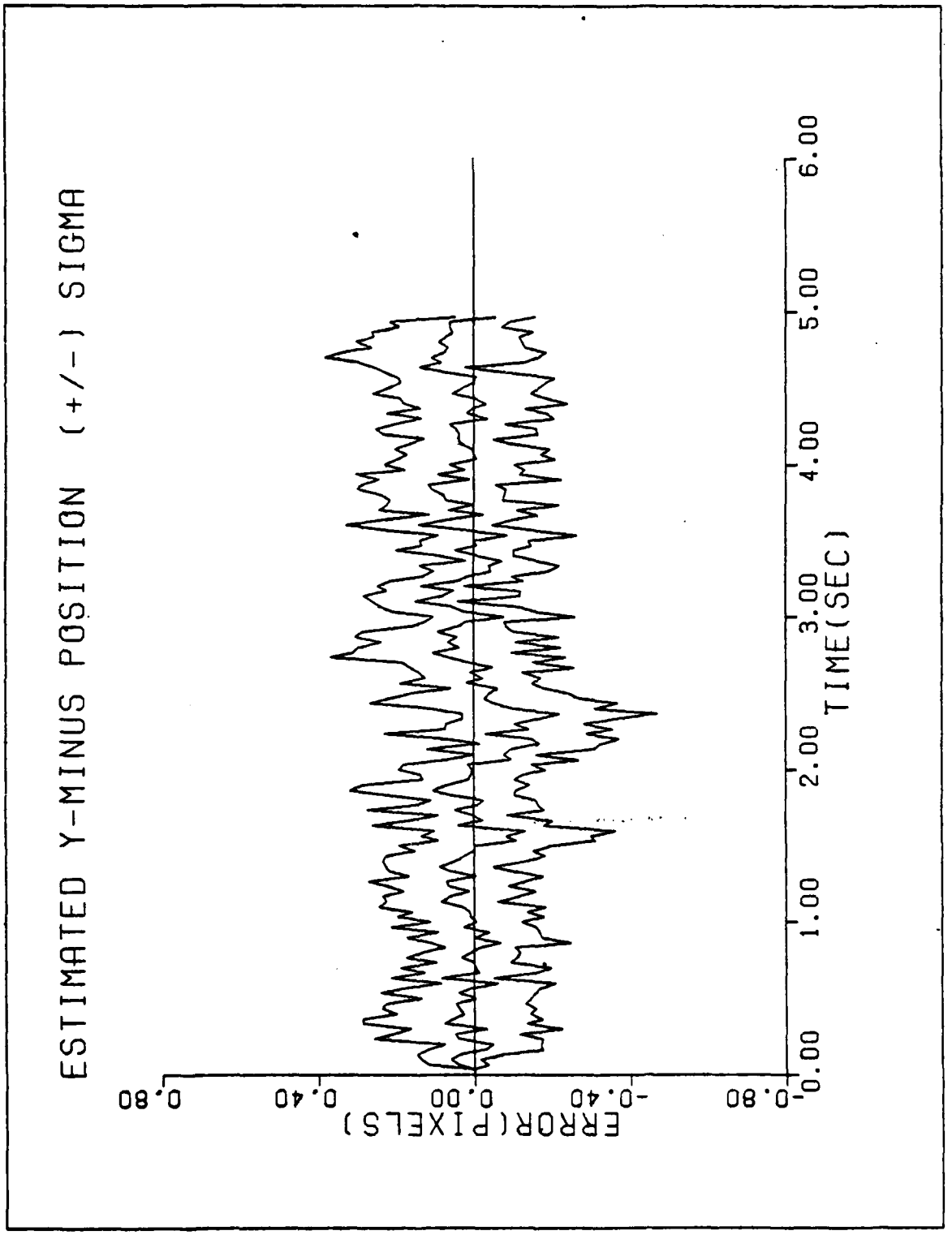


Figure C-6a. Performance Plot for C2T2F2

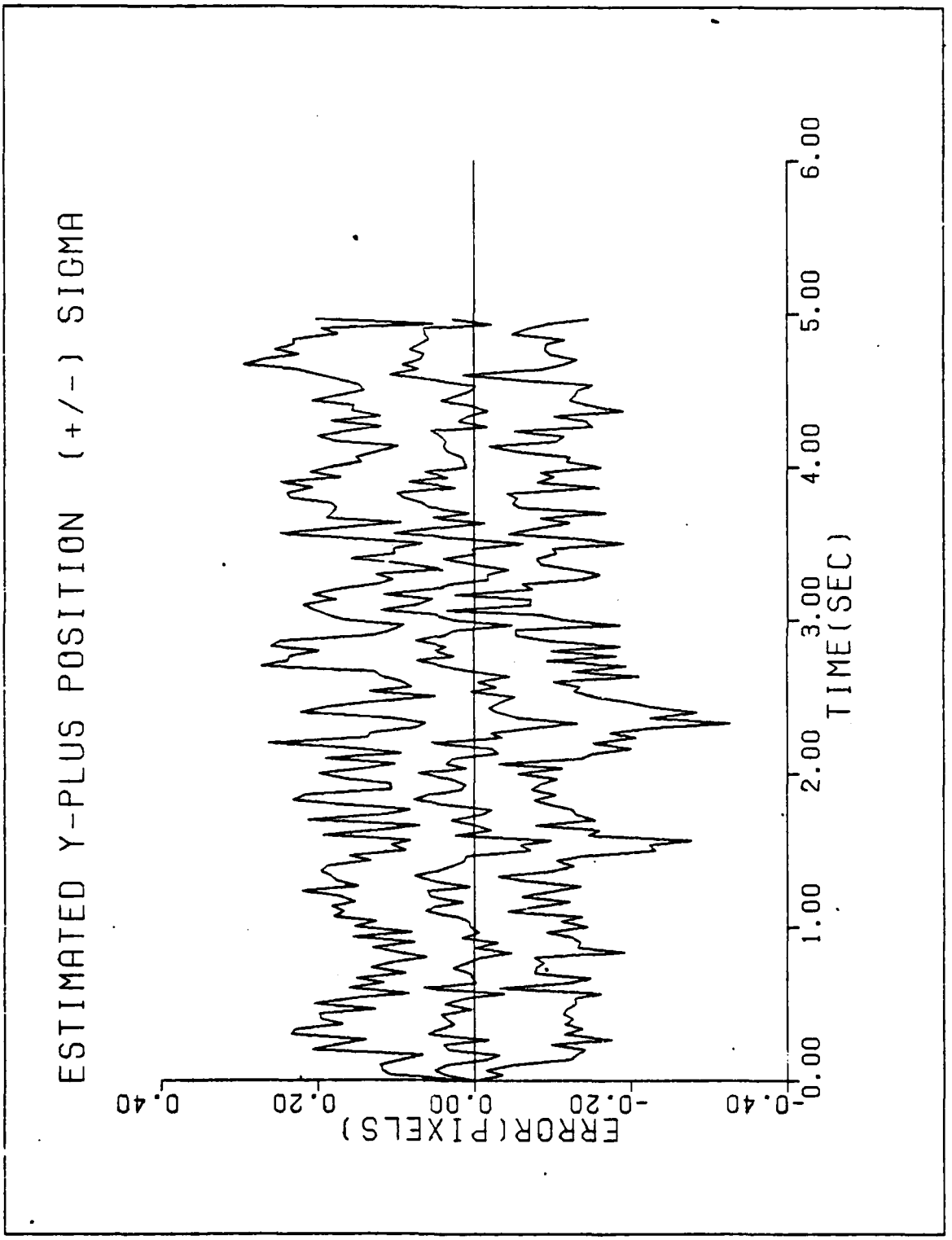


Figure C-6b. Performance Plot for C2T2F2

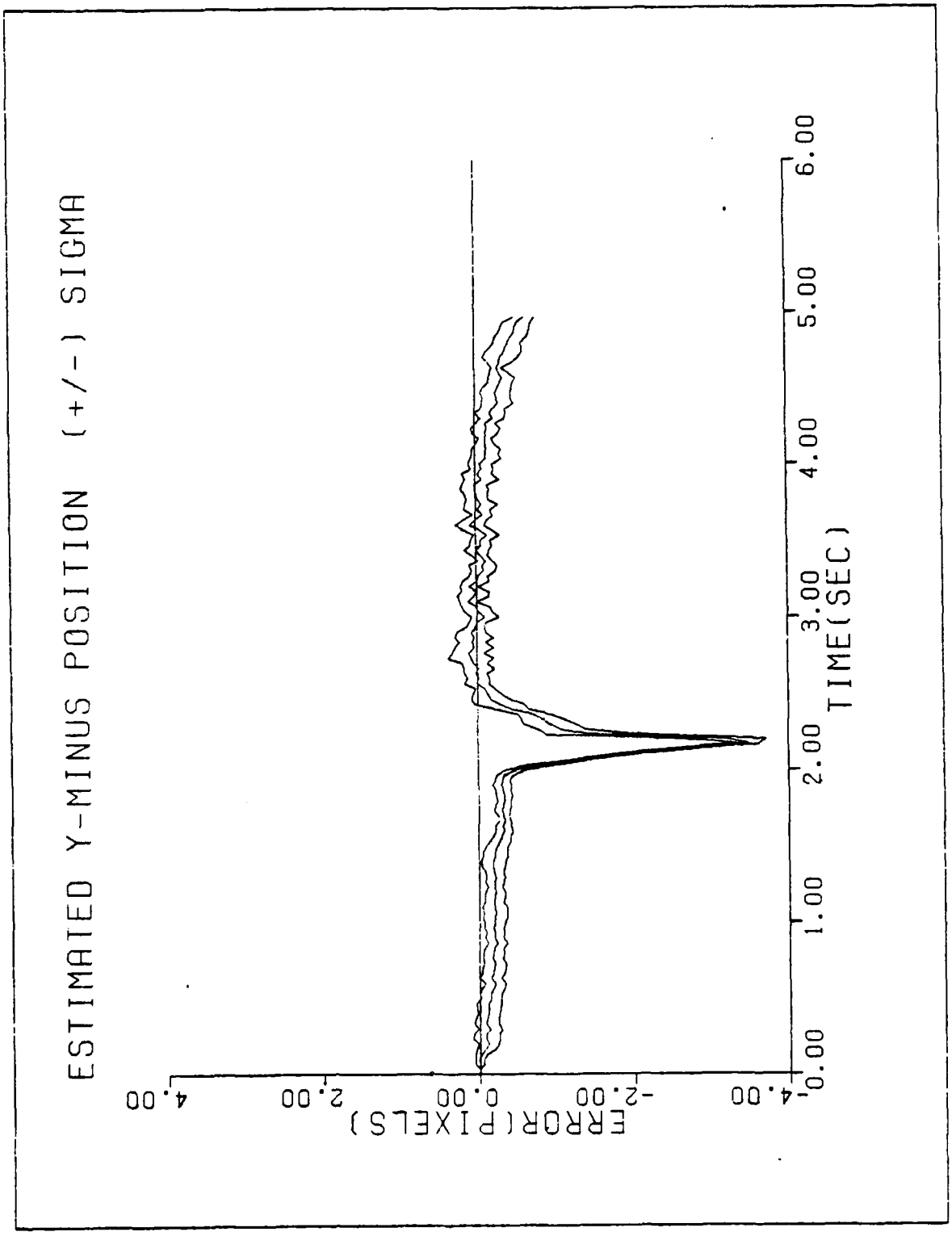


Figure C-7a. Performance Plot for C10T2MF

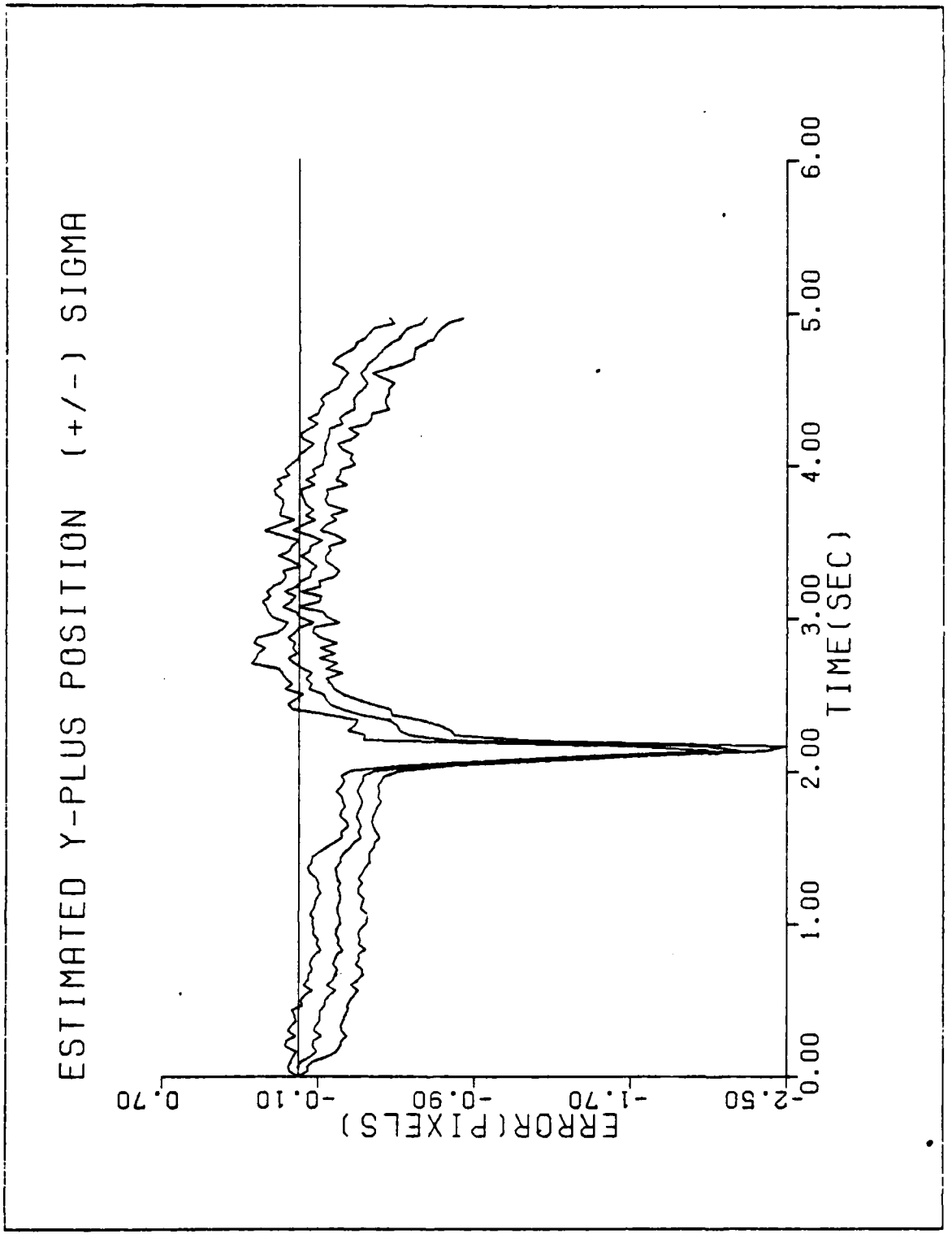


Figure C-7b. Performance Plot for C10T2MF

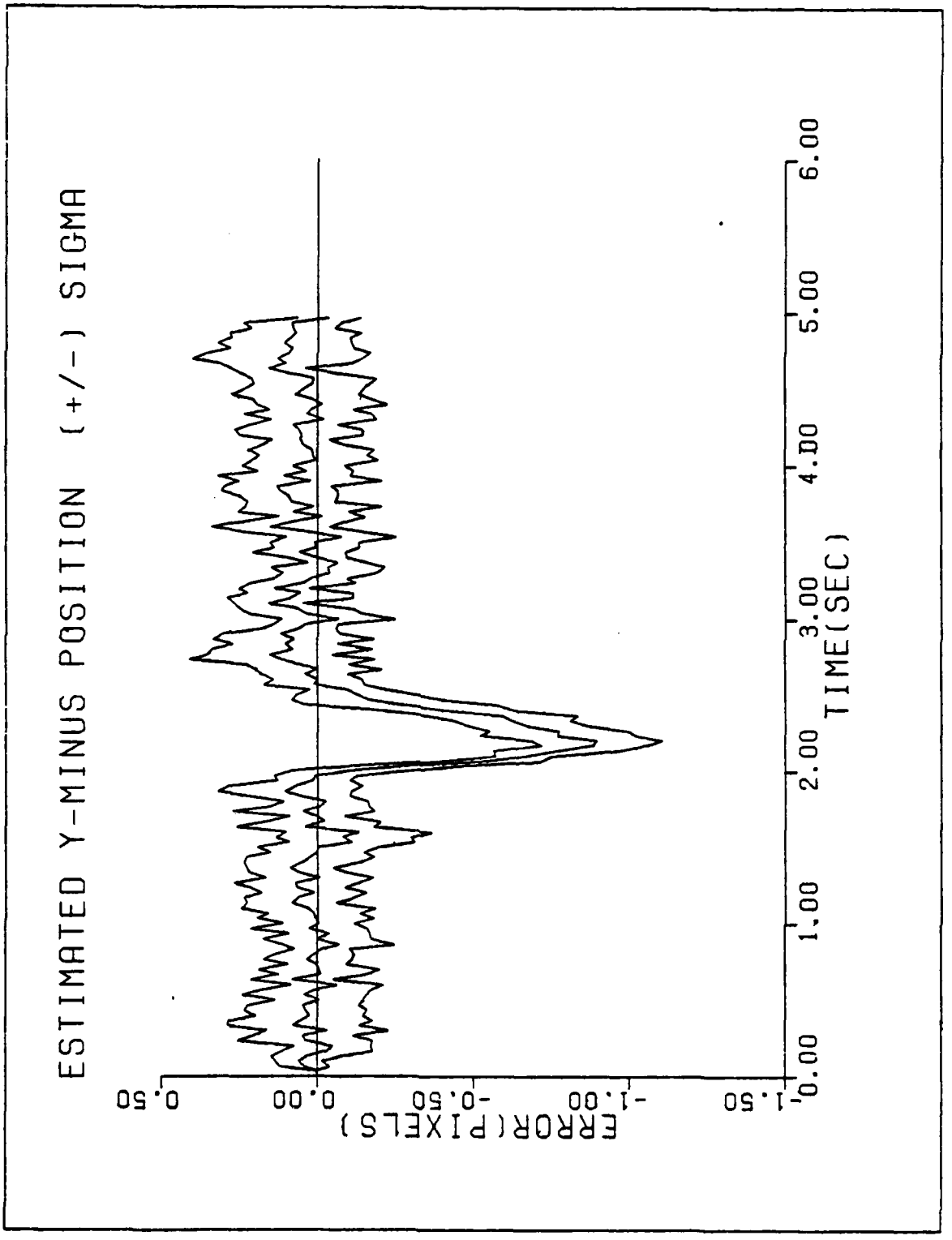


Figure C-8a. Performance Plot for C10T2F2

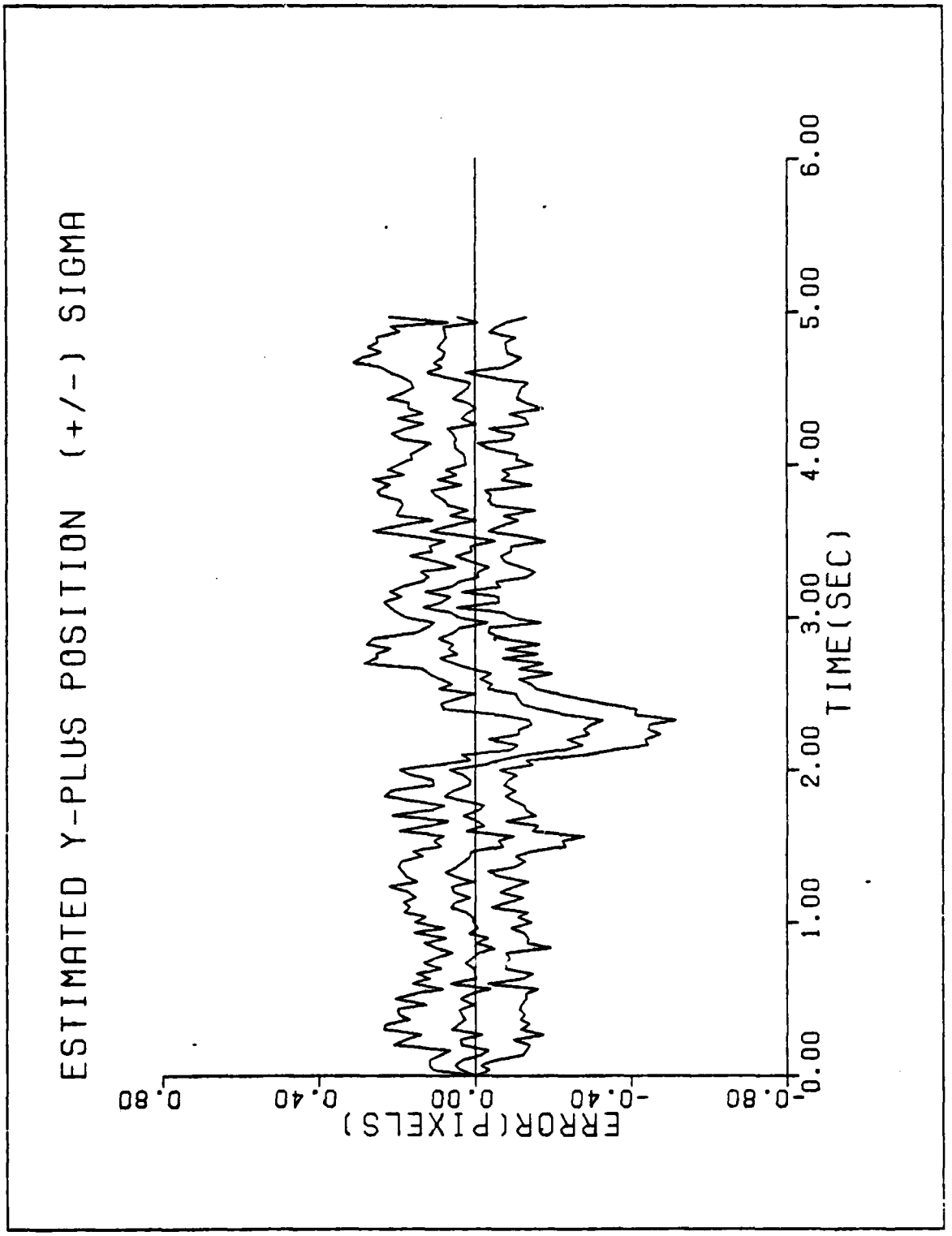


Figure C-8b. Performance Plot for C10T2F2

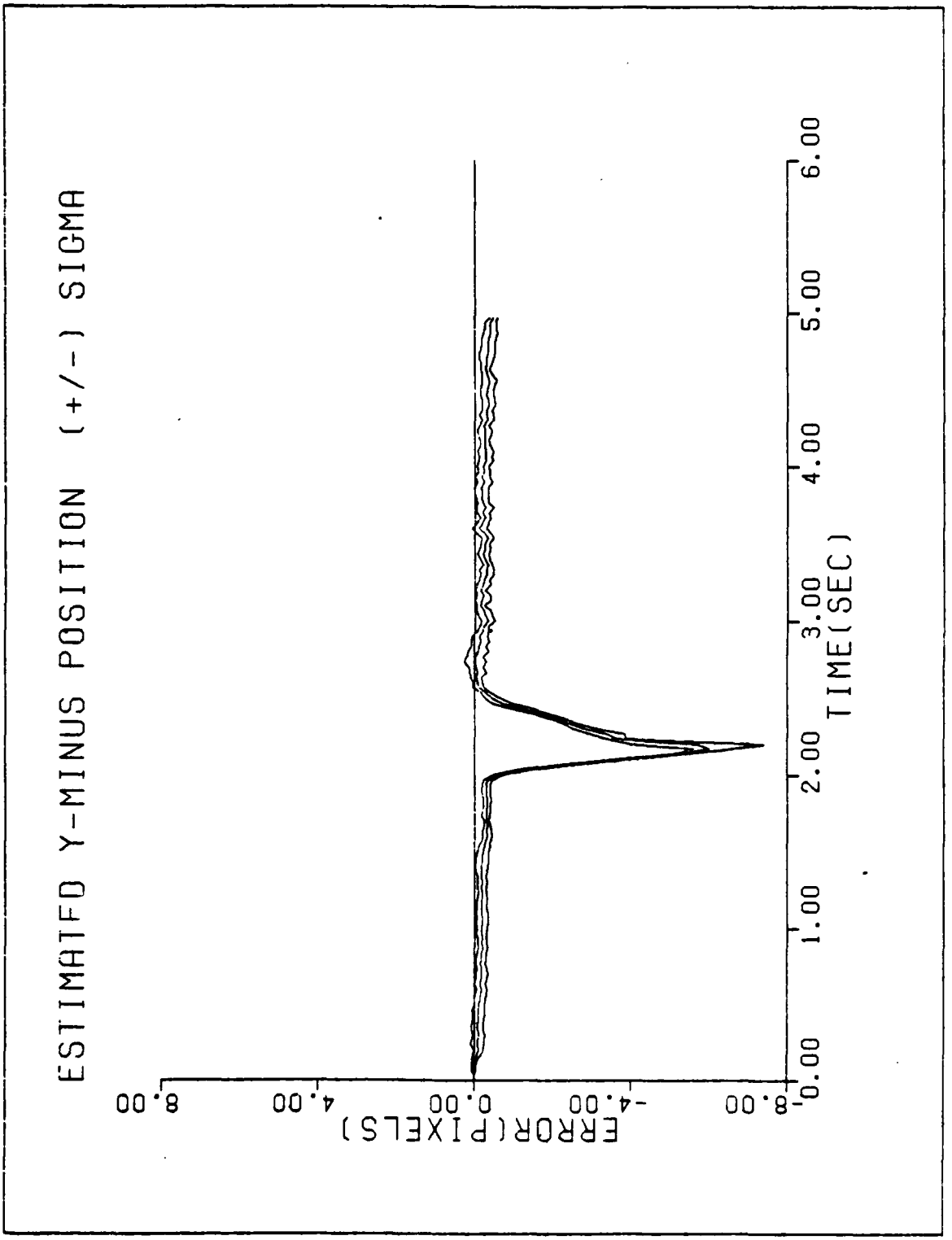


Figure C-9a. Performance Plot for C20T2MF

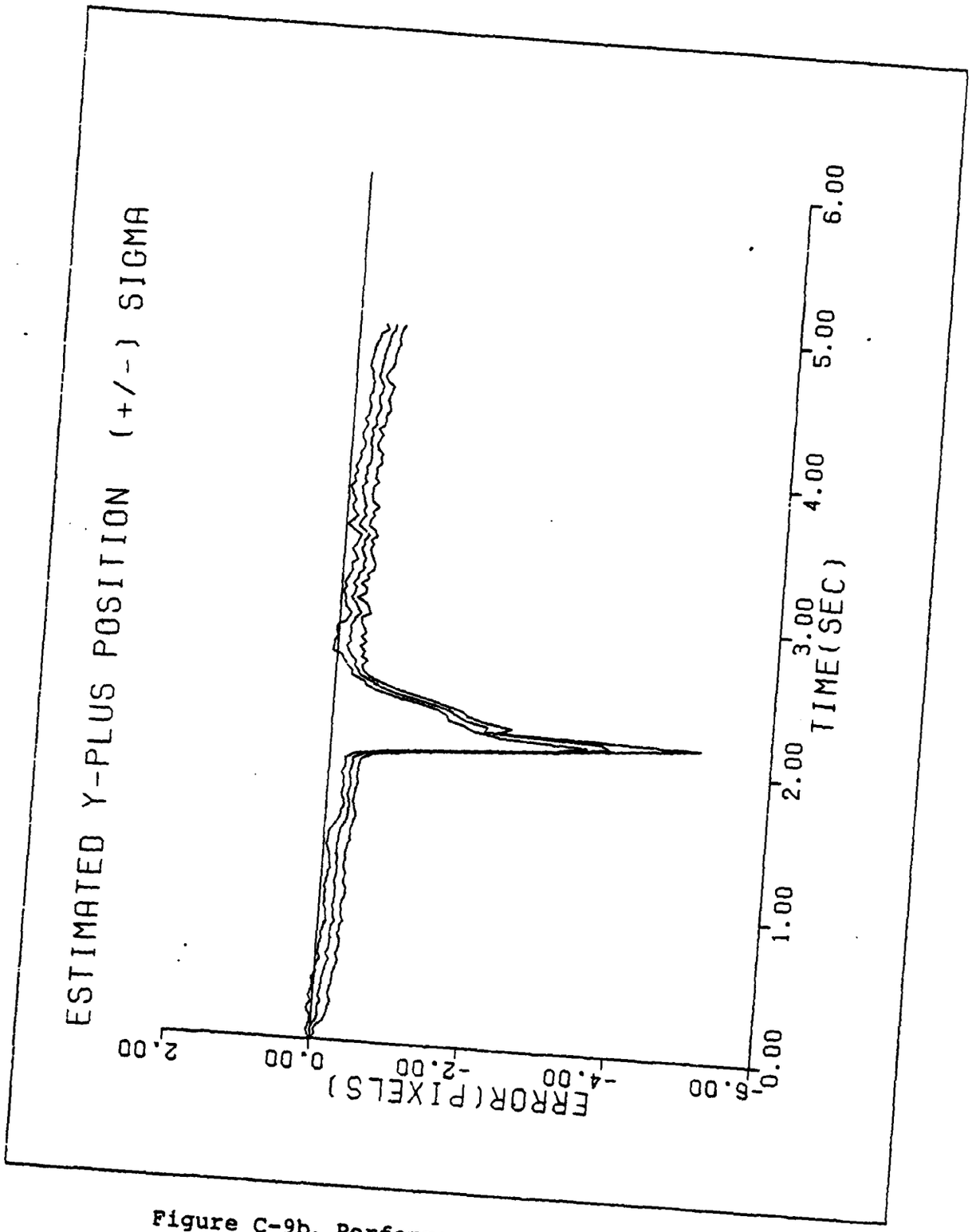


Figure C-9b. Performance Plot for C20T2MF

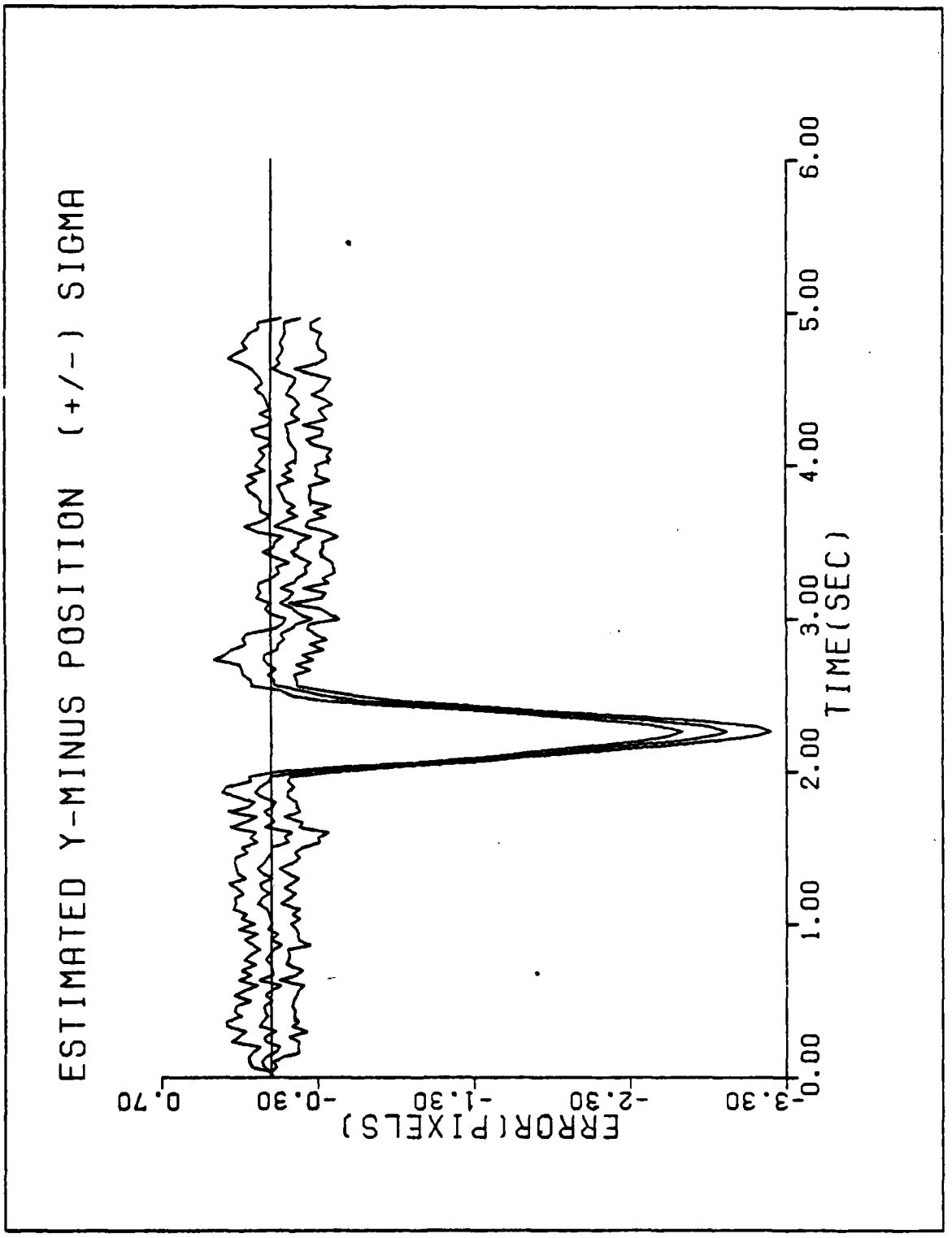


Figure C-10a. Performance Plot for C20T2F2

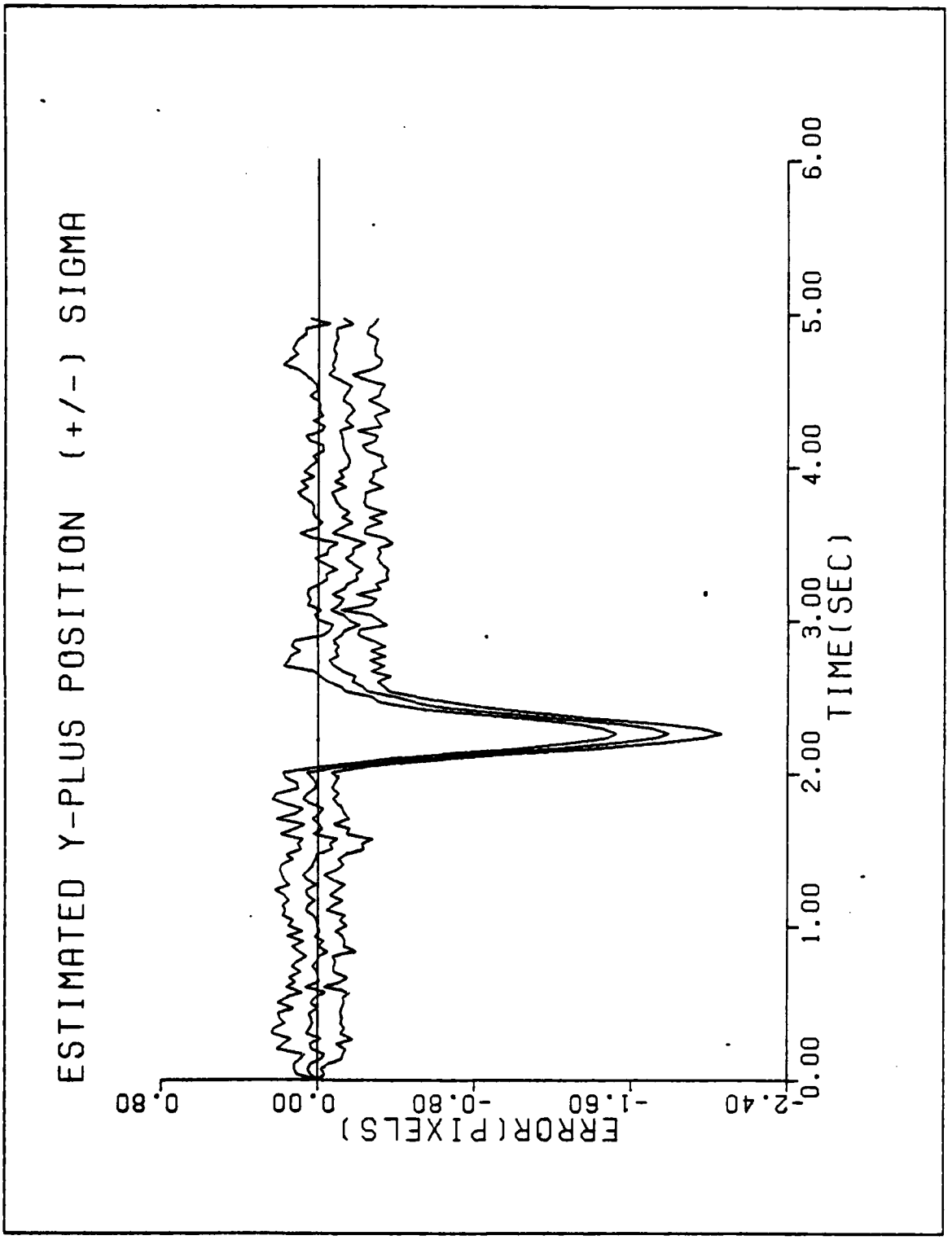


Figure C-10b. Performance Plot for C20T2F2

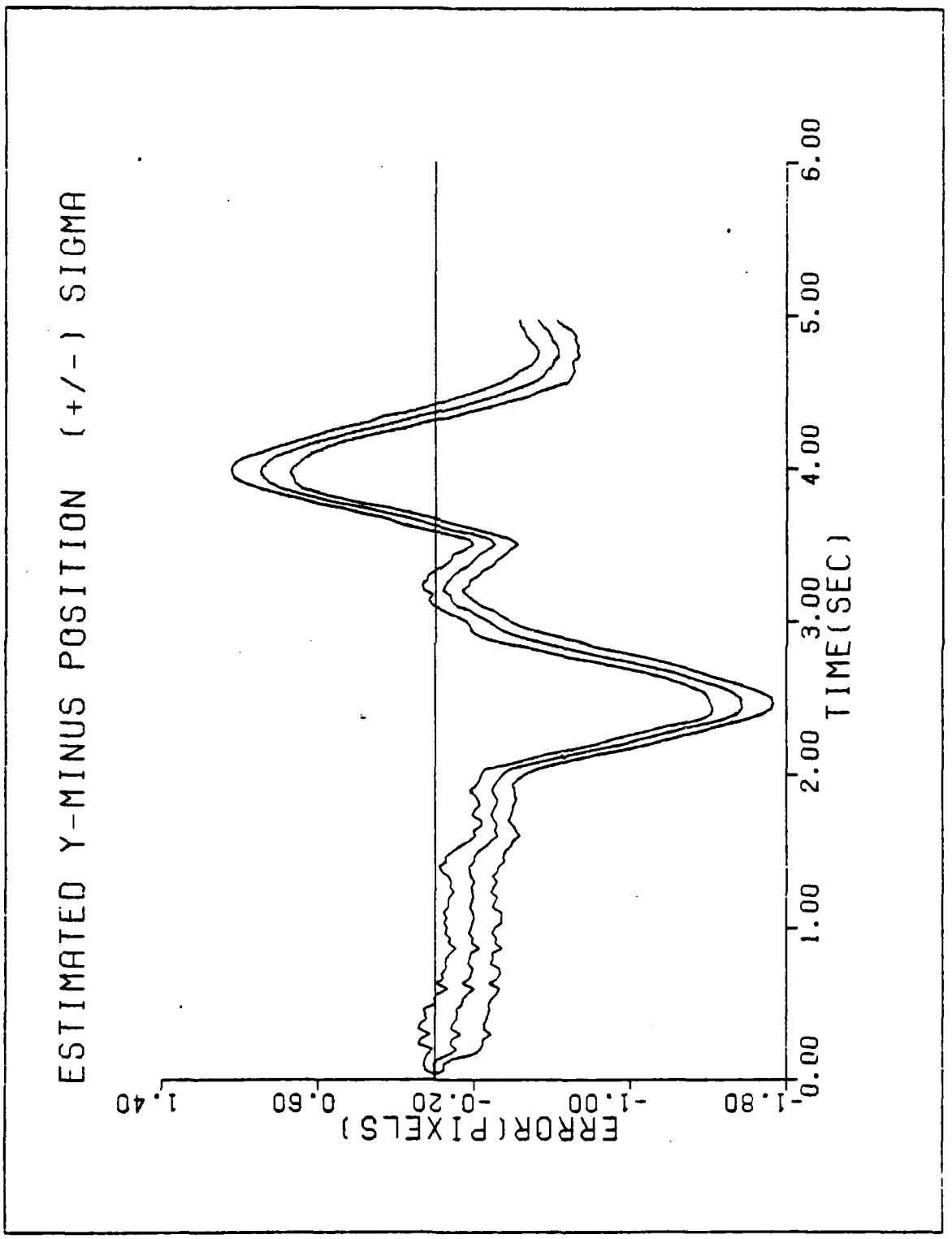


Figure C-11a. Performance Plot for C2T3MF

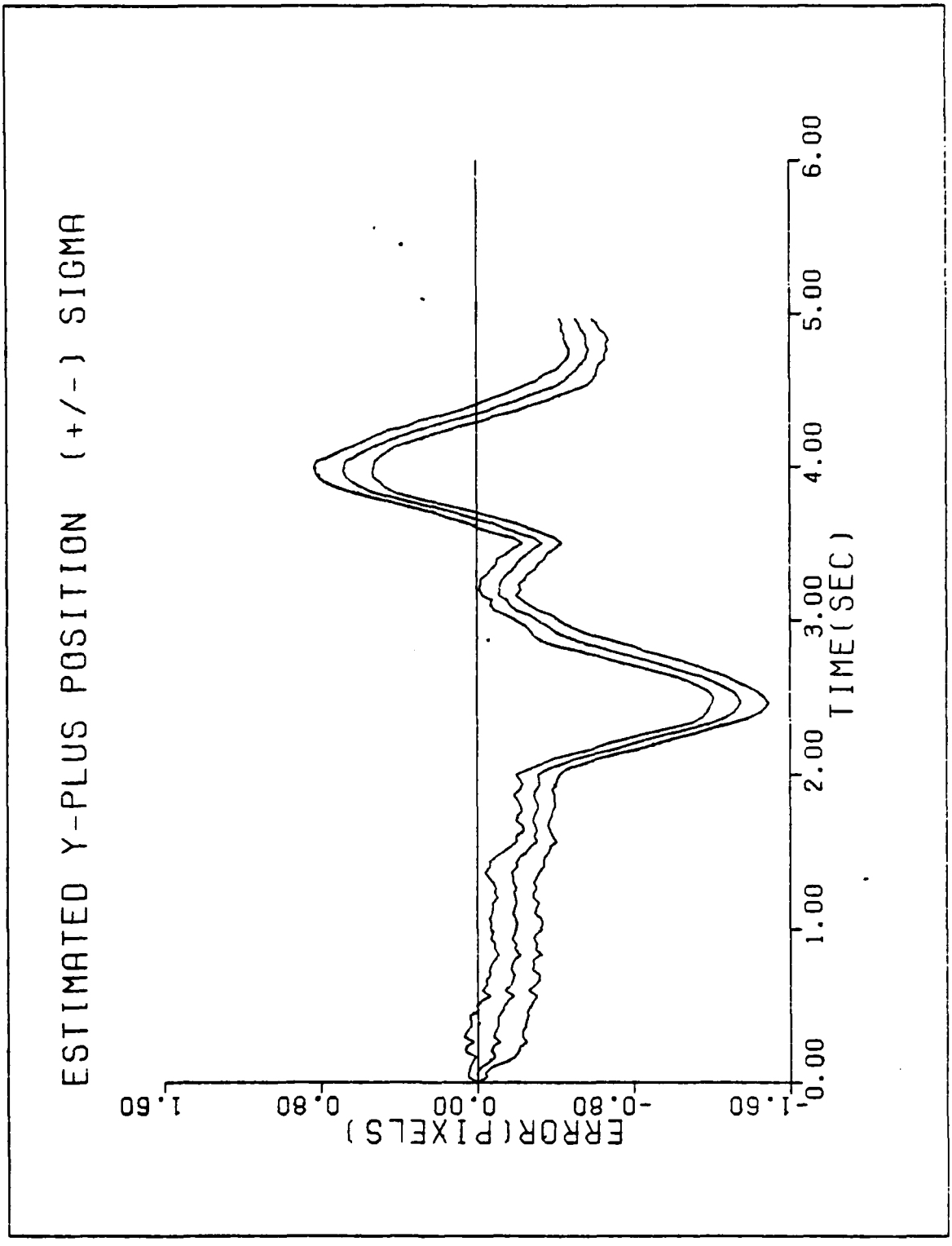


Figure C-11b. Performance Plot for C2T3MF

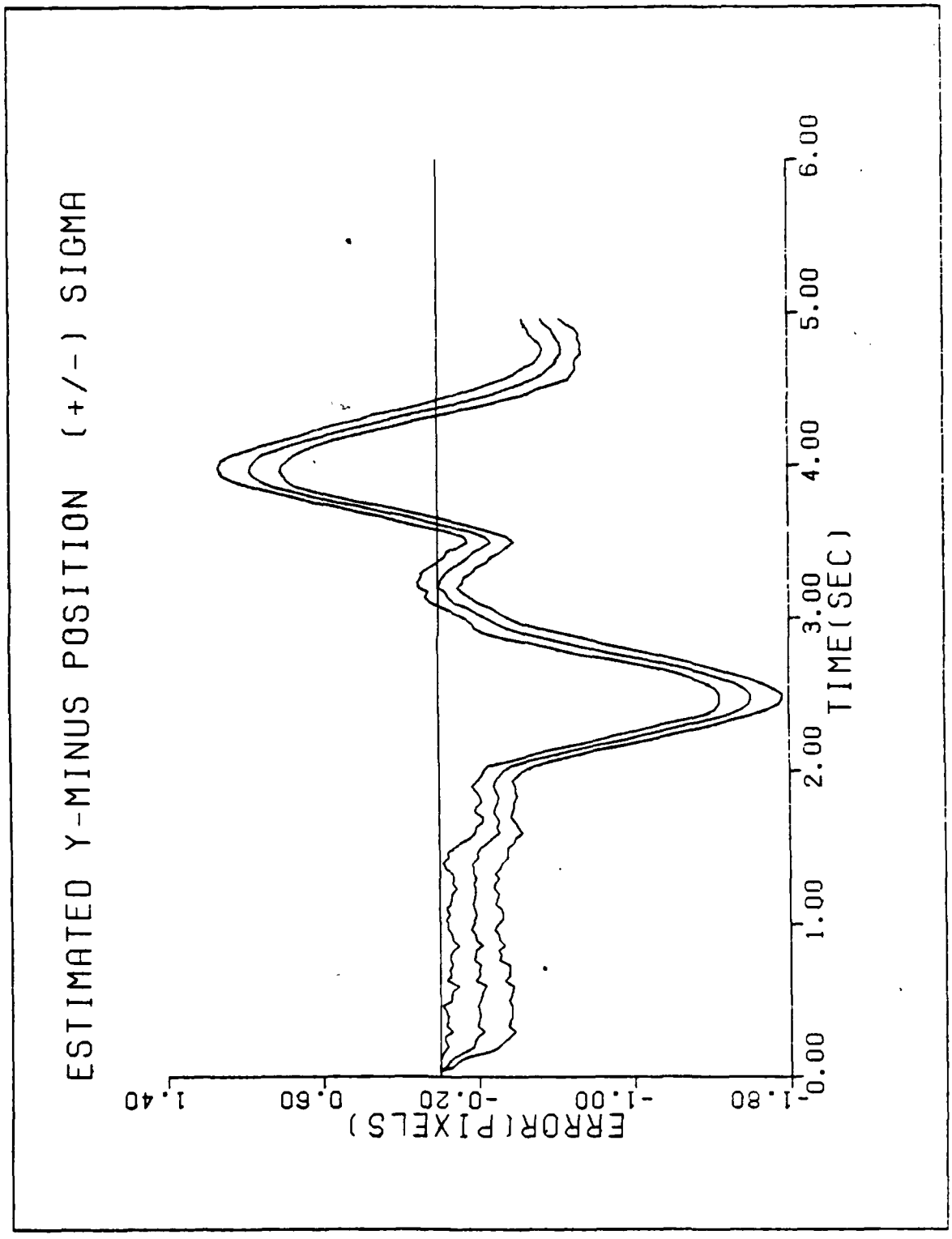


Figure C-12a. Performance Plot for C2T3F1

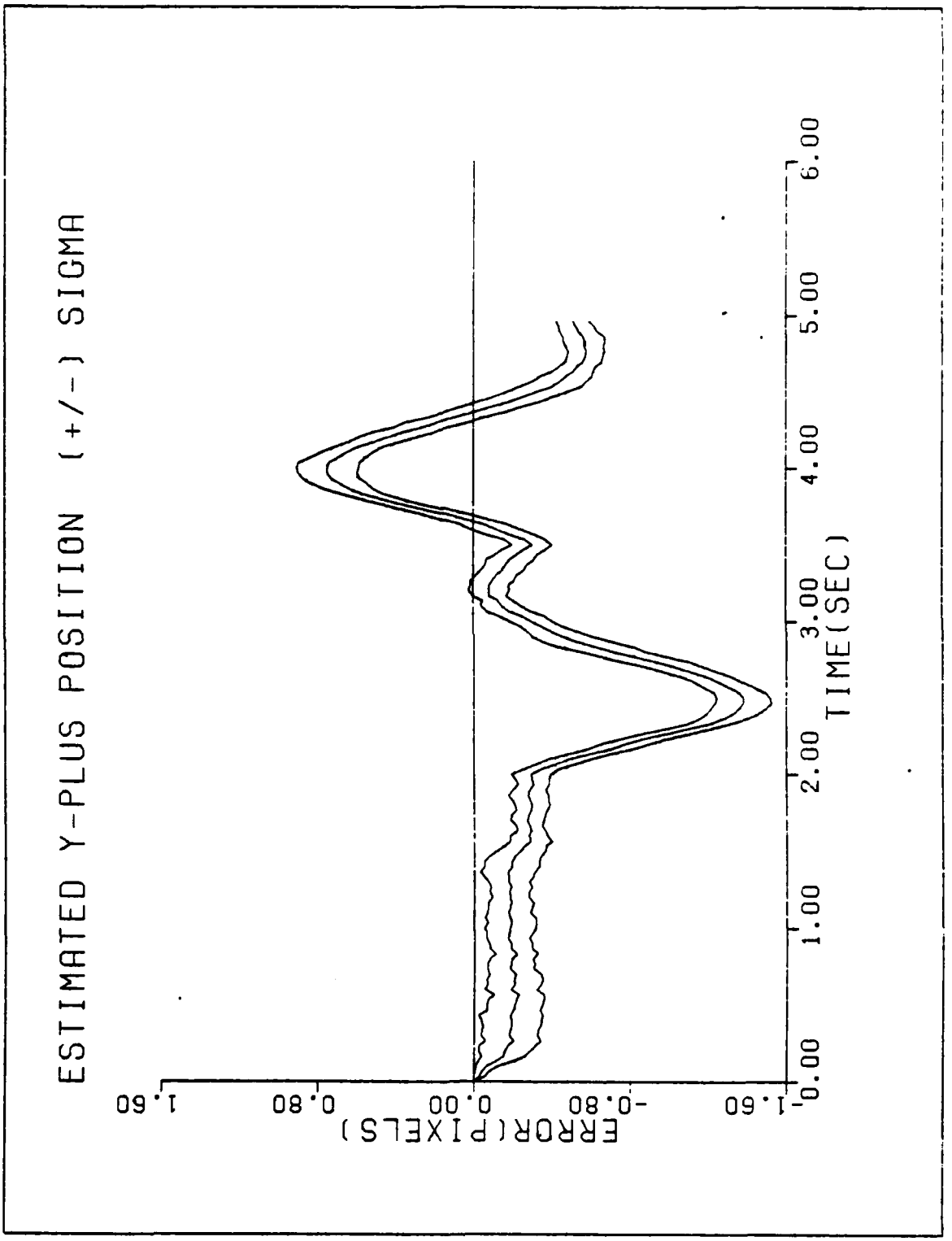


Figure C-12b. Performance Plot for C2T3F1

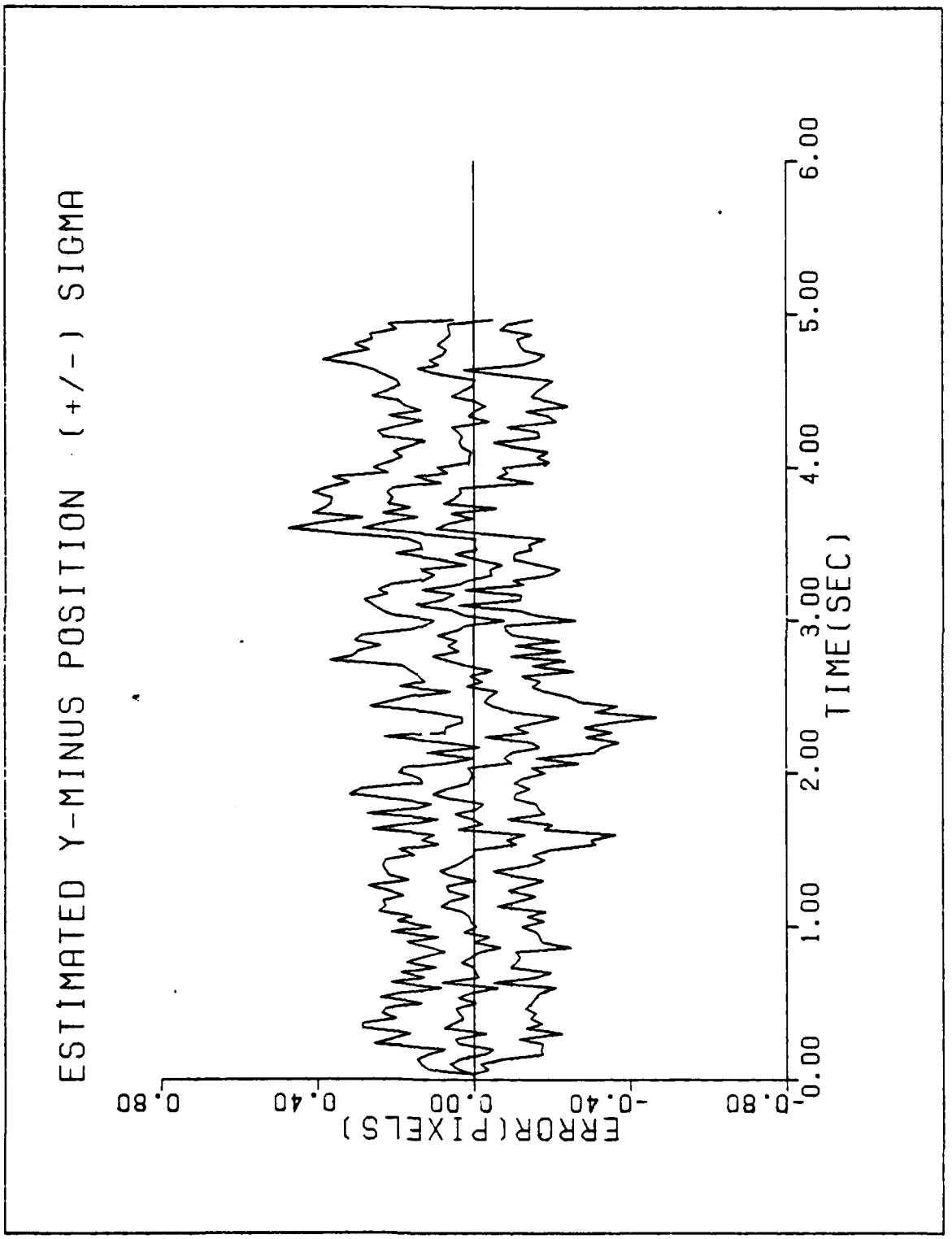


Figure C-13a. Performance Plot for C2T3F2

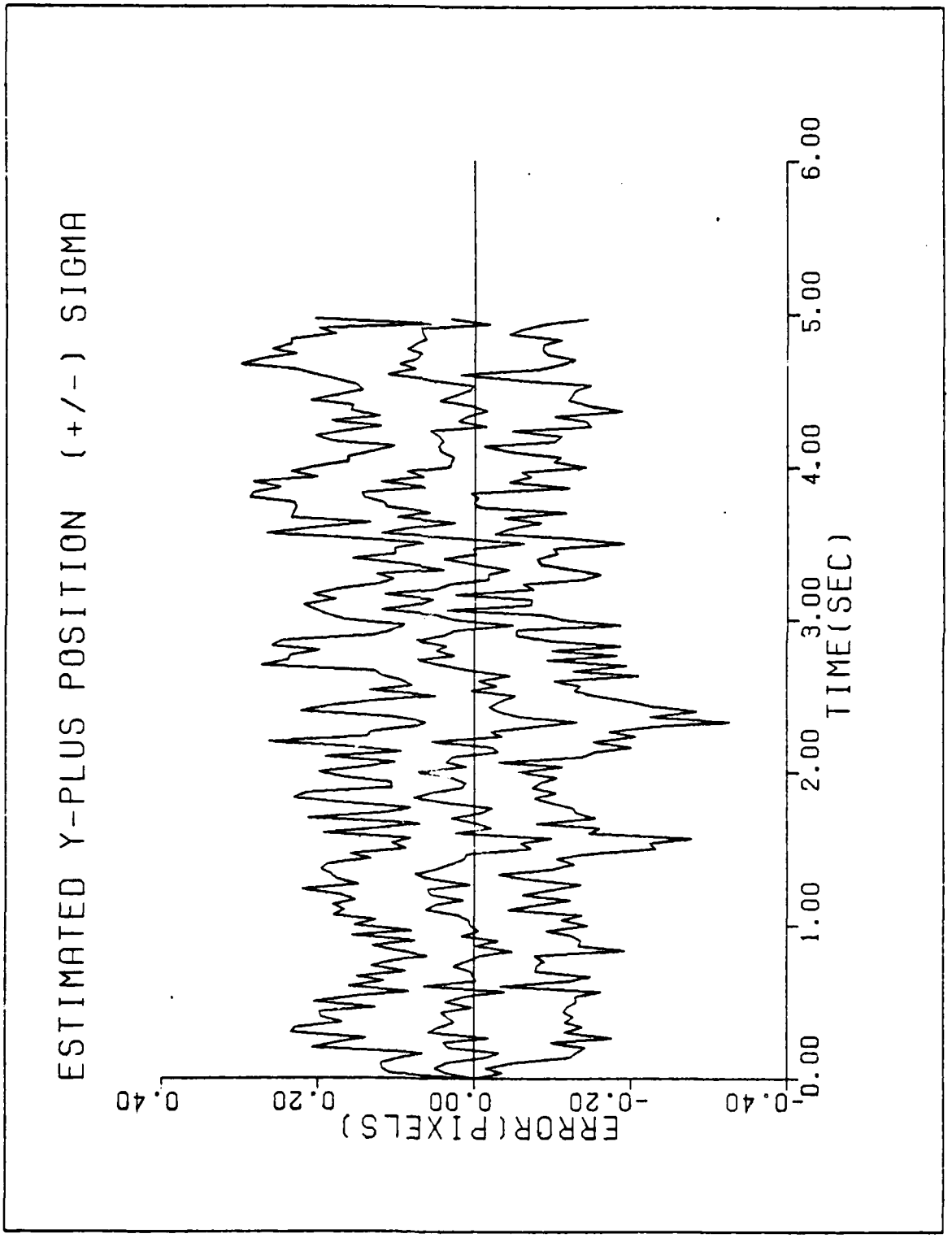


Figure C-13b. Performance Plot for C2T3F2

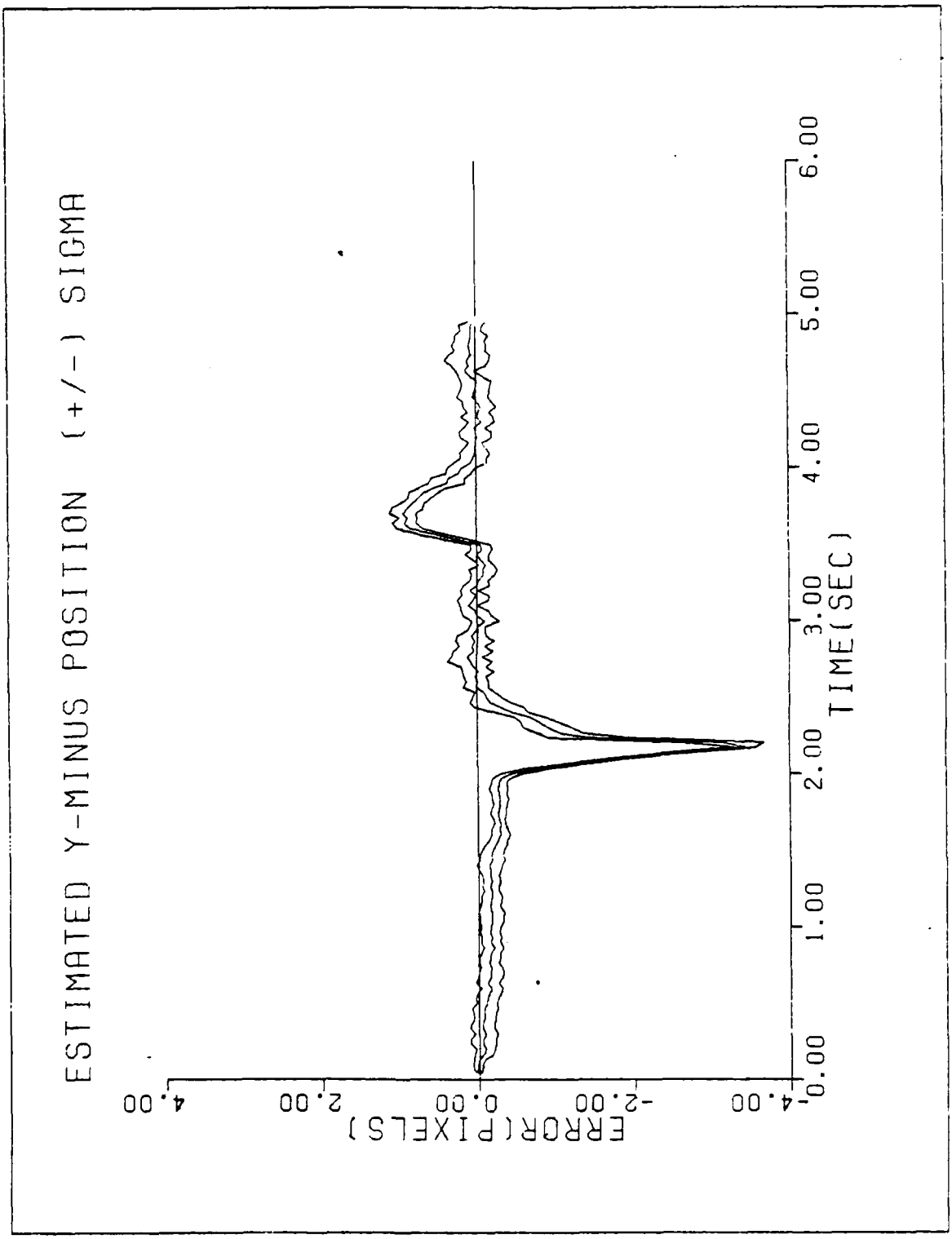


Figure C-14a. Performance Plot for C10T3MF

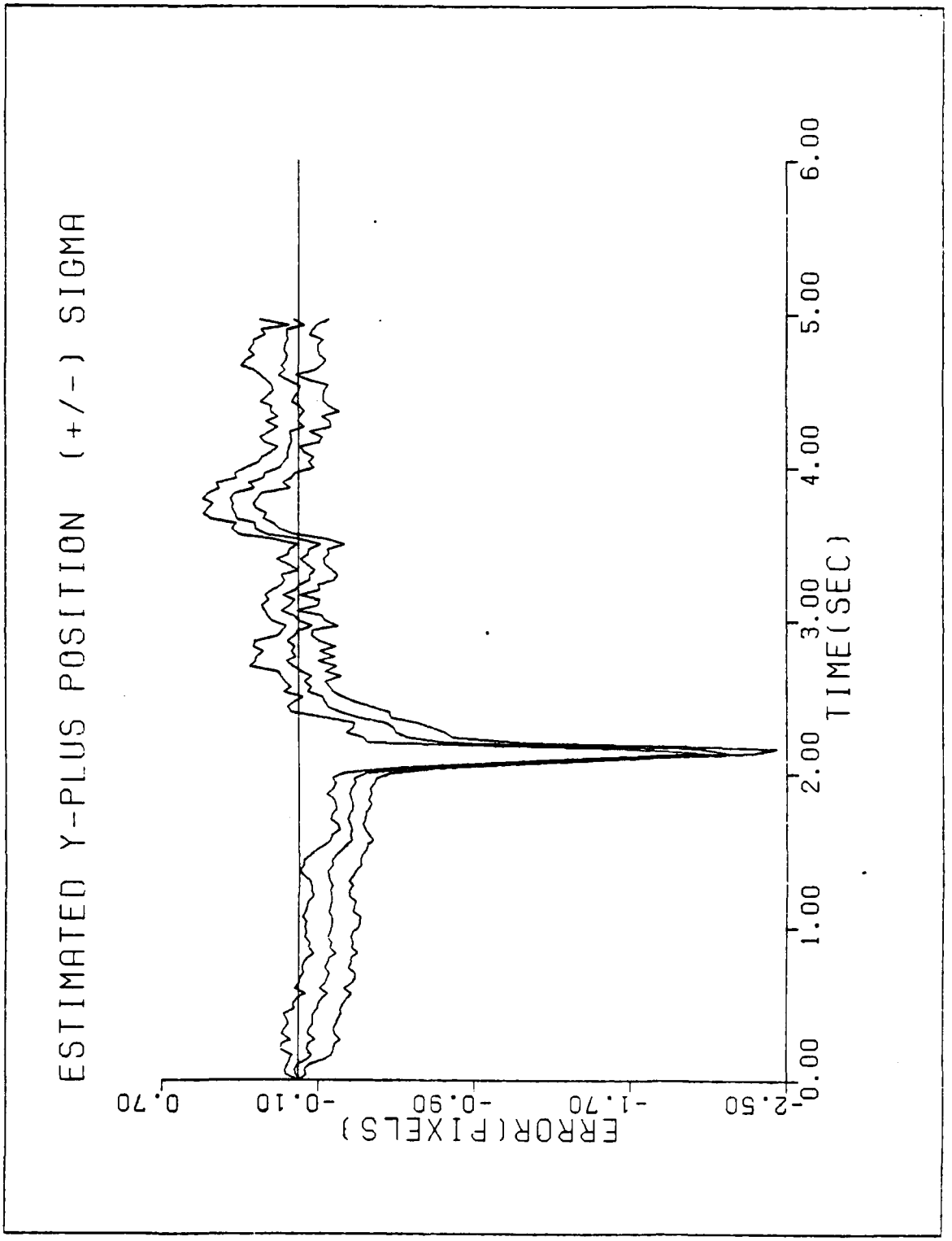


Figure C-14b. Performance Plot for C10T3MF

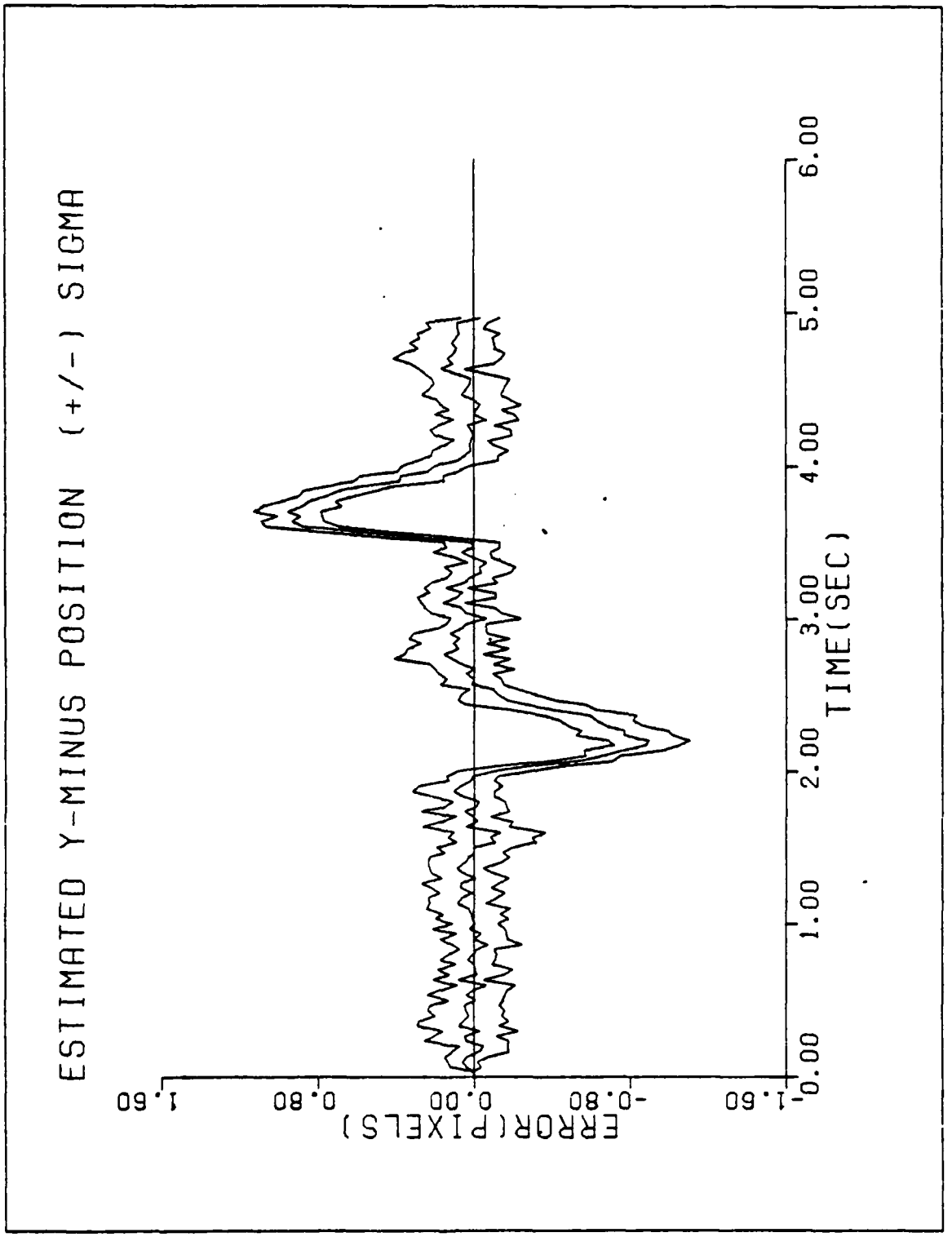


Figure C-15a. Performance Plot for C10T3F2

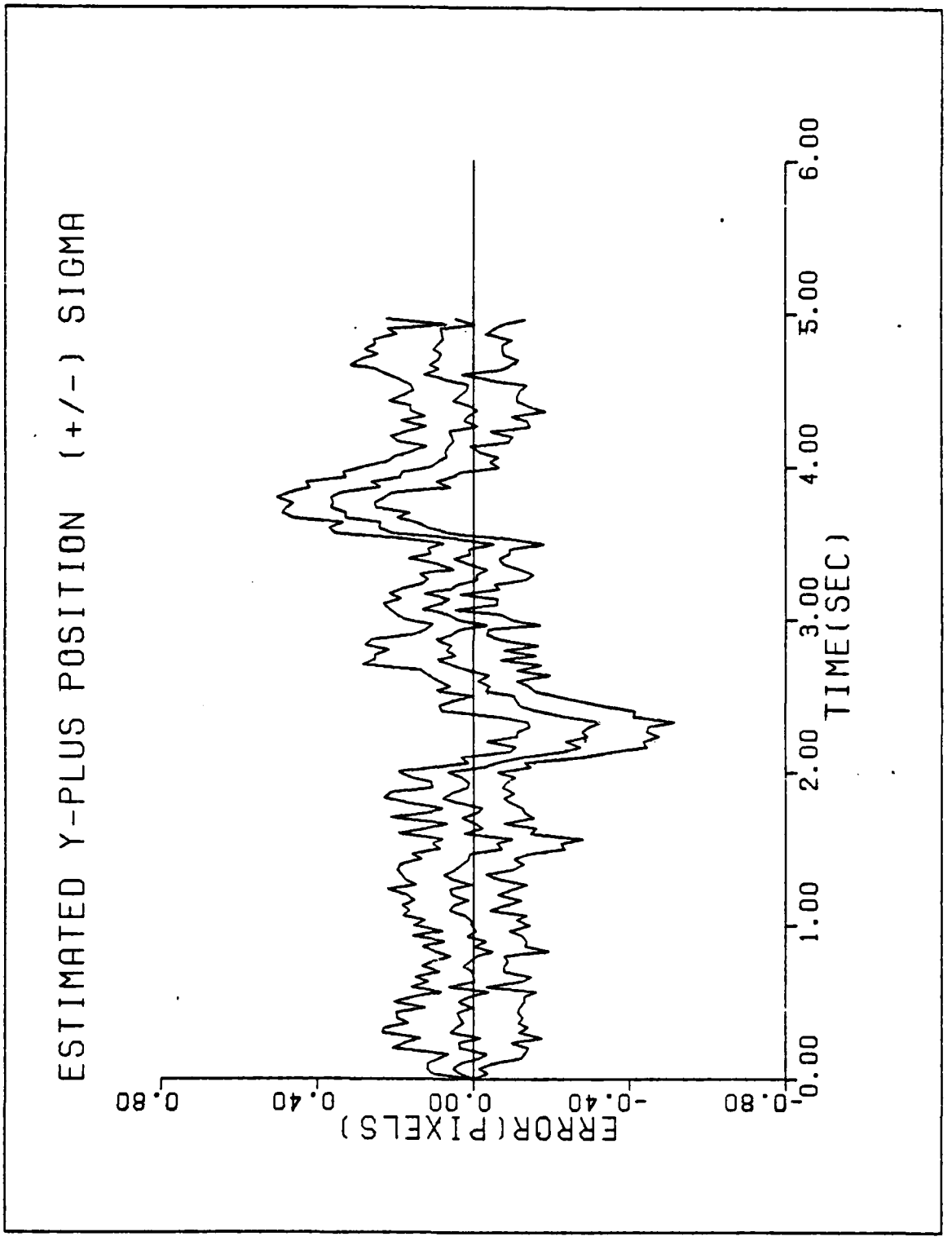


Figure C-15b. Performance Plot for C10T3F2

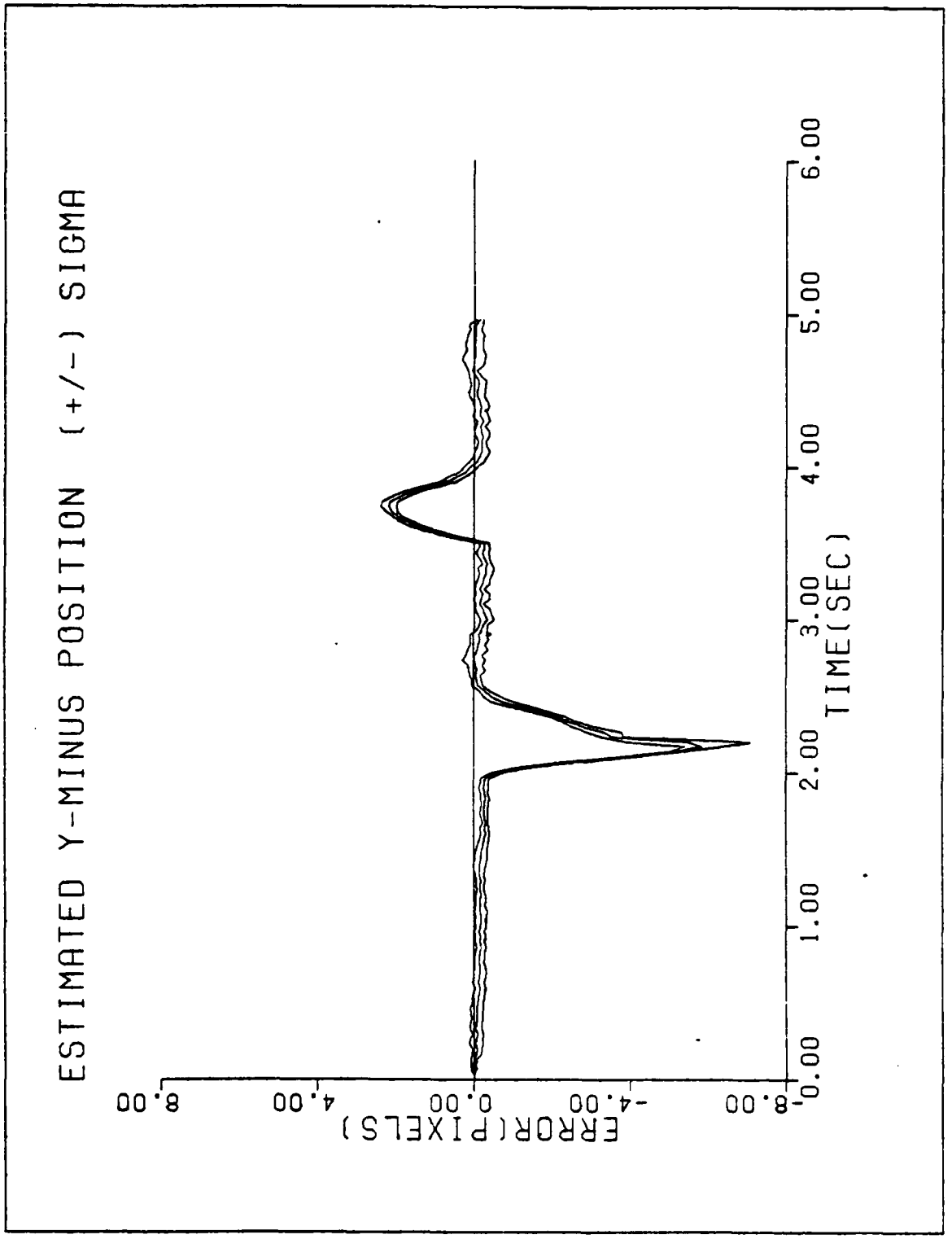


Figure C-16a. Performance Plot for C20T3MF

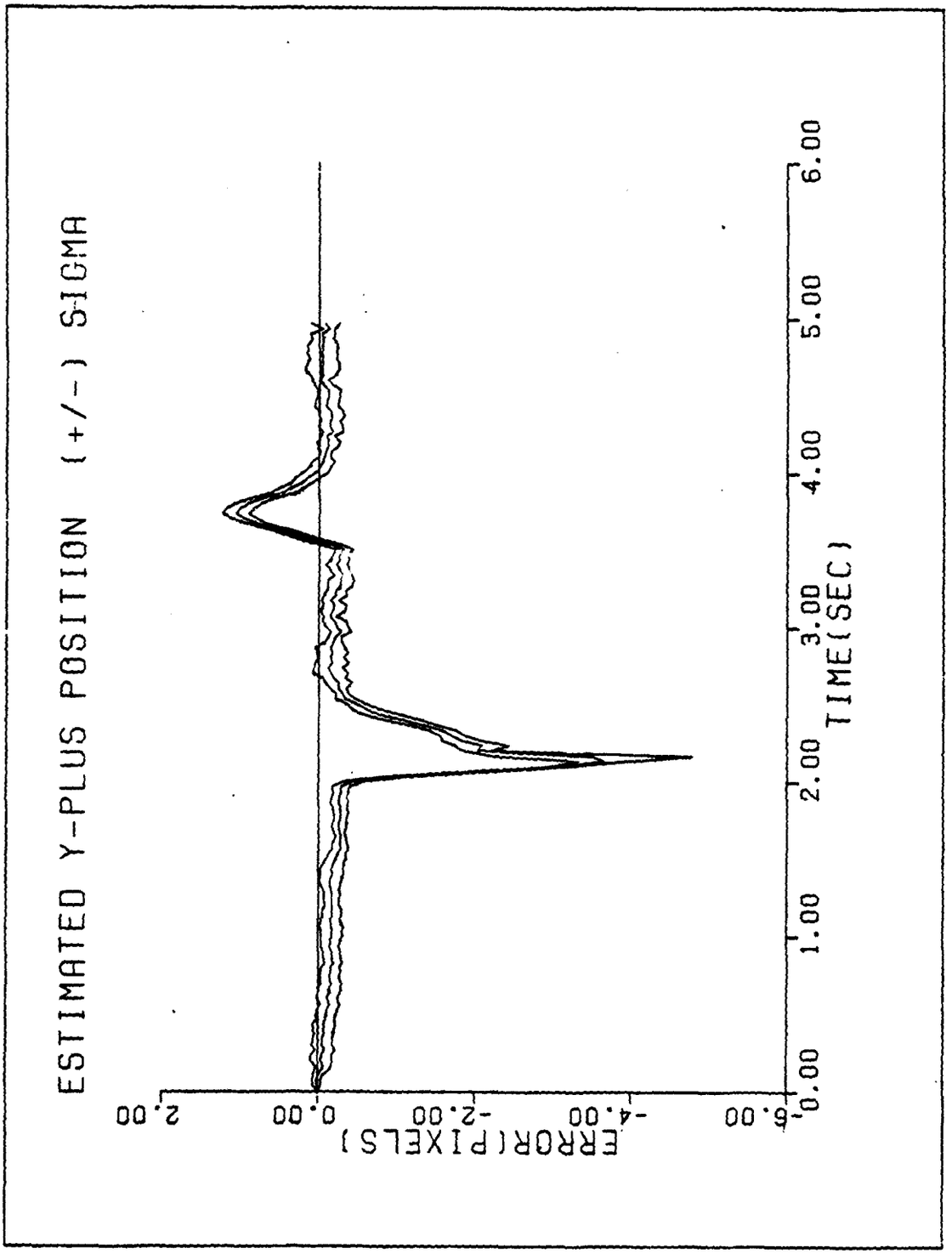


Figure C-16b. Performance Plot for C20T3MF

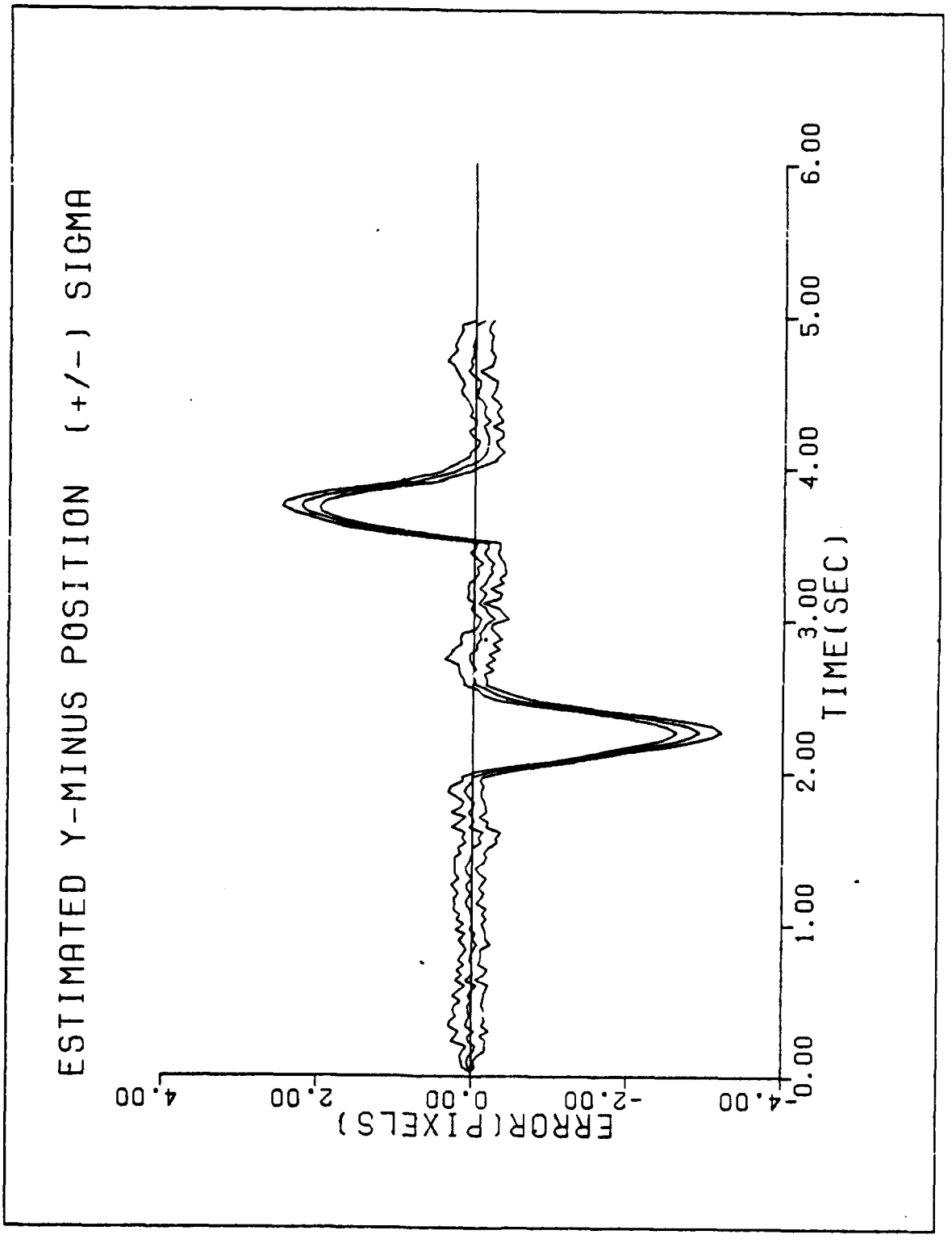


Figure C-17a. Performance Plot for C20T3F2

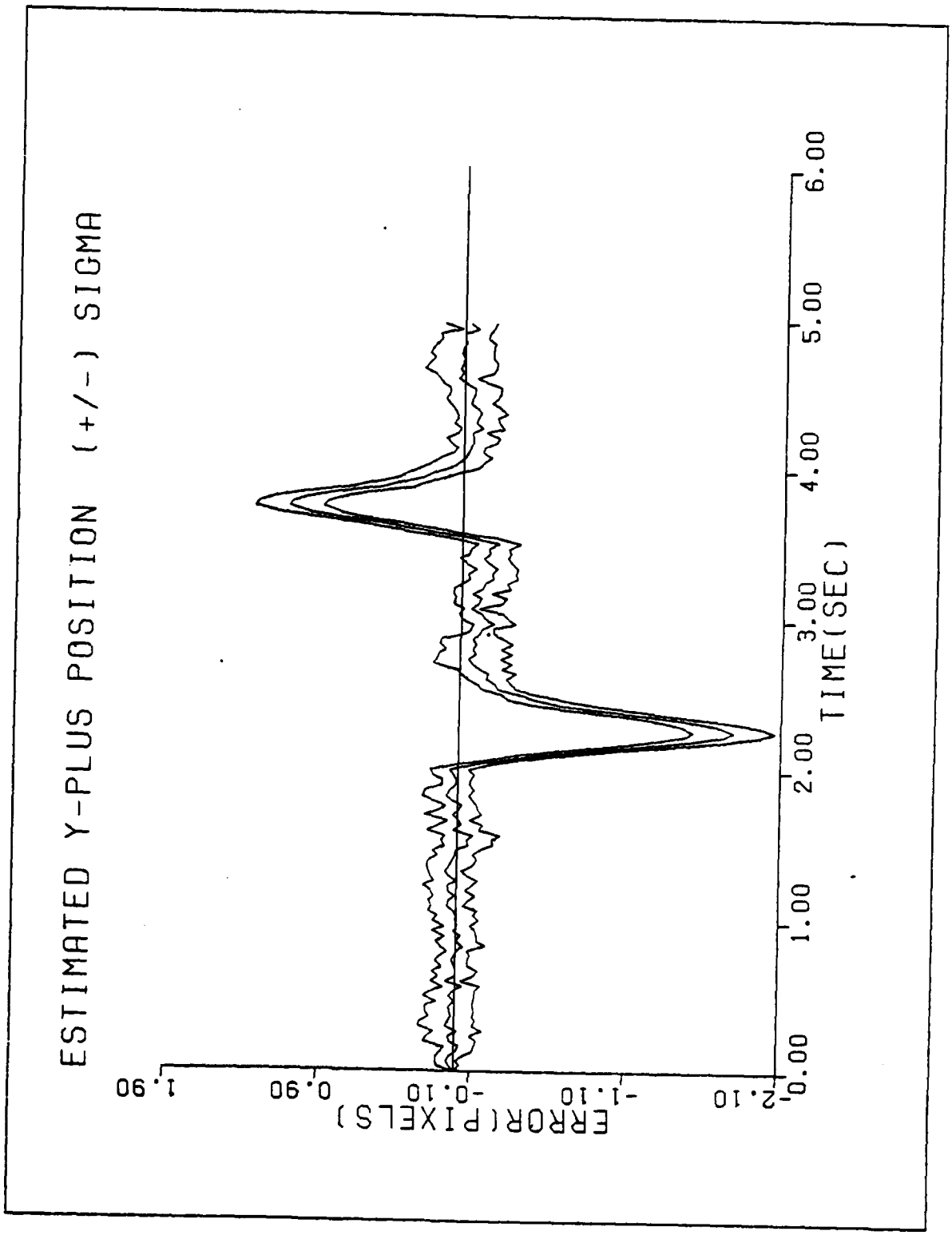


Figure C-17b. Performance Plot for C20T3F2

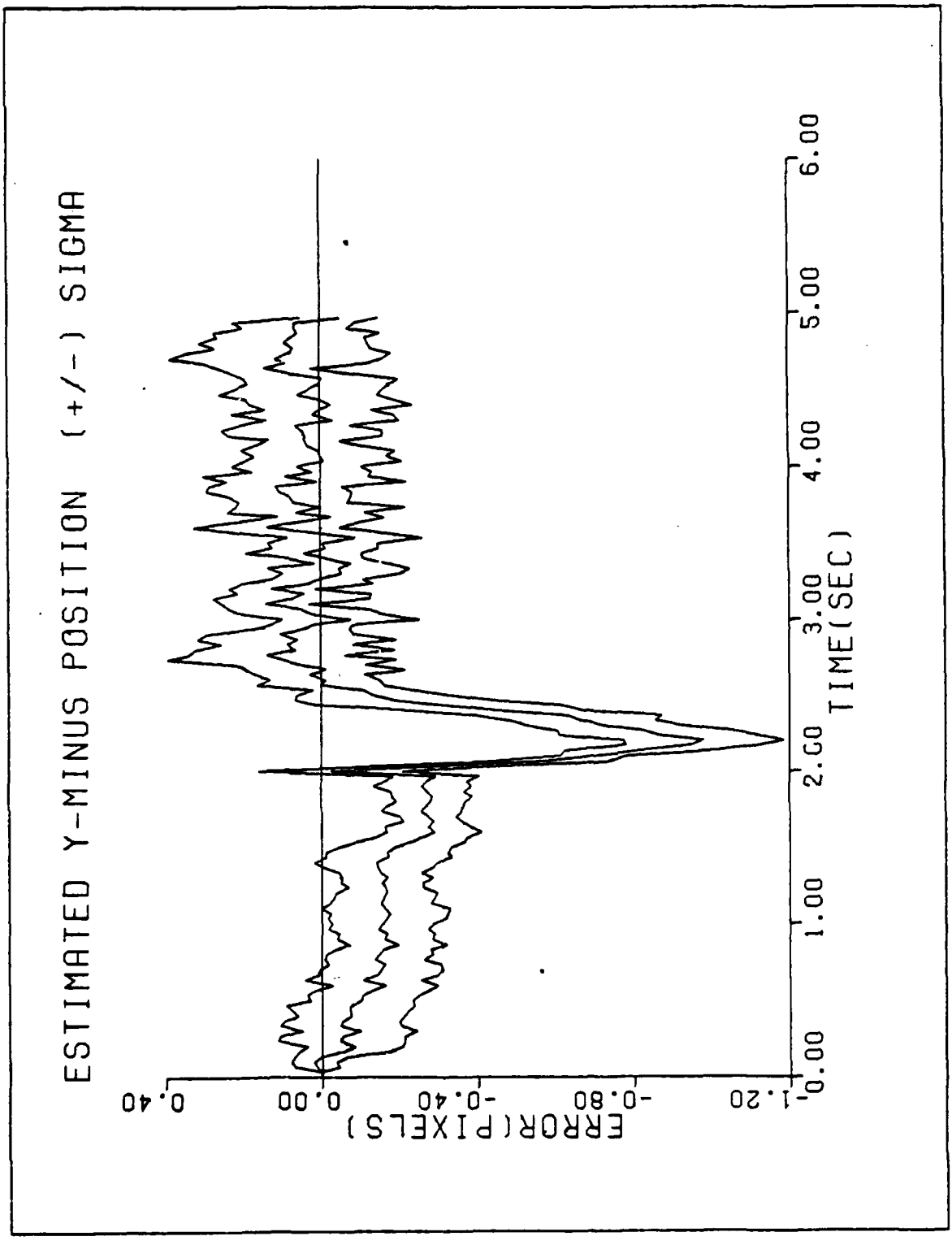


Figure C-18a. Performance Plot for CA10T2MF

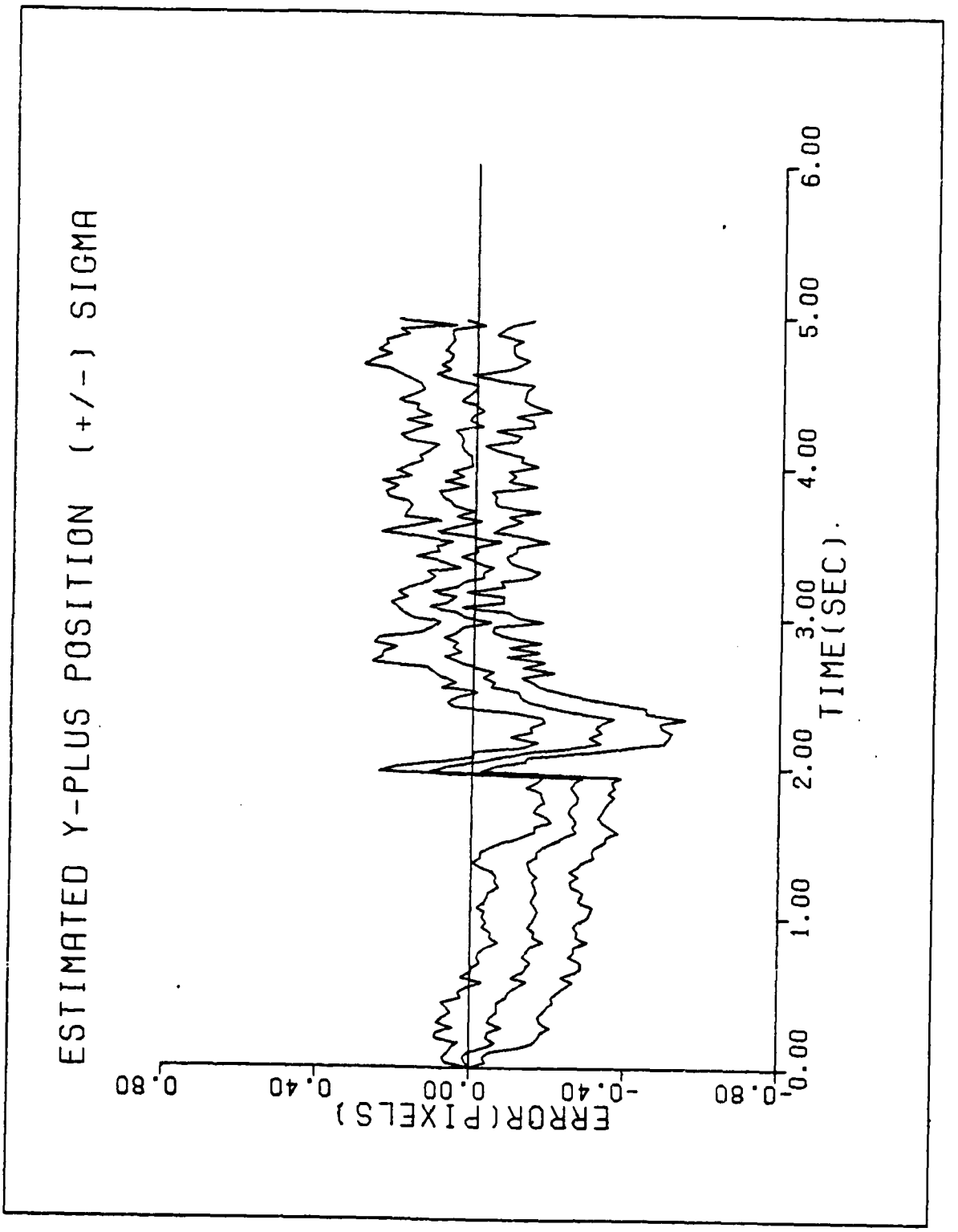


Figure C-18b. Performance Plot for CA10T2MF

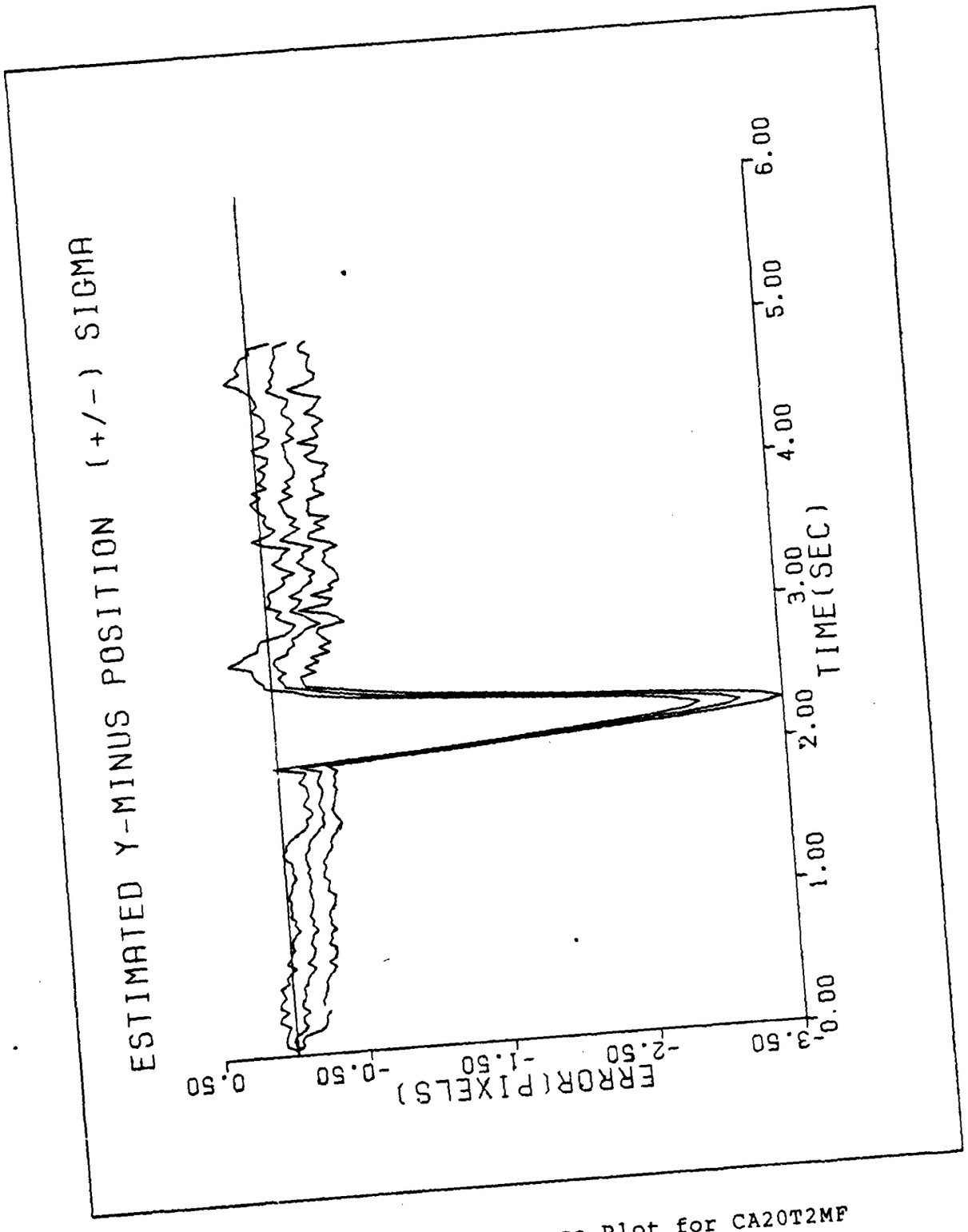


Figure C-19a. Performance Plot for CA20T2MF

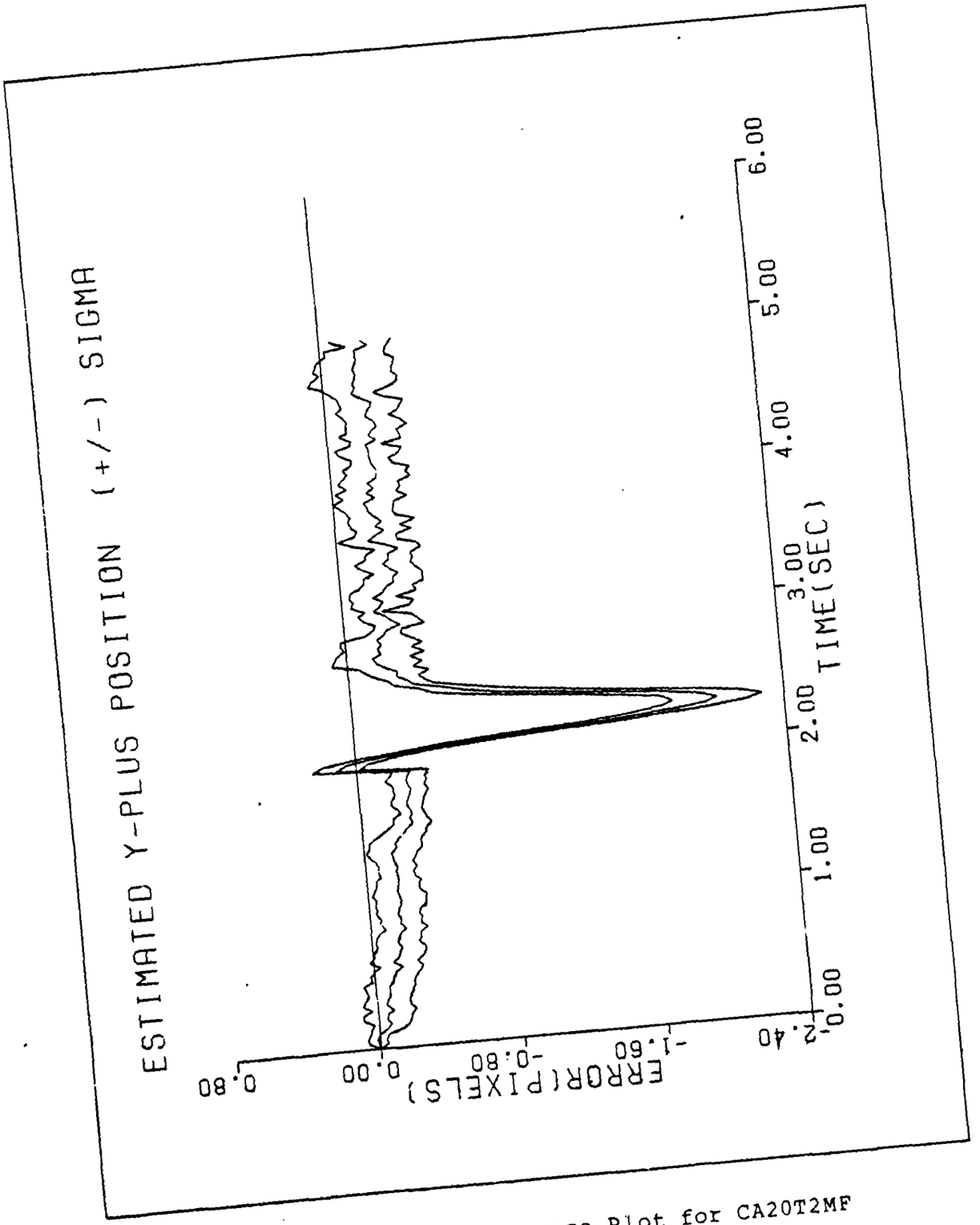


Figure C-19b. Performance Plot for CA20T2MF

Vita

Robert I. Suizu was born on January 14, 1959 in Honolulu, Hawaii. He graduated from Maui High School in May 1976. In May 1980, he graduated from the United States Air Force Academy with a Bachelor of Science degree in Electrical Engineering. From June 1980 to June 1982, he served as an electrical engineer at the Aeronautical Systems Division Computer Center. In June 1982, Lt. Suizu entered the Air Force Institute of Technology to pursue a Master of Science Degree in Electrical Engineering. Lt. Suizu is a member of Tau Beta Pi.

Permanent address: 176 Ehilani St
Pukalani, HI 96788

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A141 144

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GE/EE/84M-4	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: School of Engineering; 6b. OFFICE SYMBOL (If applicable): AFIT/EN	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code): Air Force Institute of Technology, Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Air Force Weapons Lab; 8b. OFFICE SYMBOL (If applicable): AFWL/ARAA	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code): Kirtland AFB, New Mexico 87117	10. SOURCE OF FUNDING NOS. — PROGRAM ELEMENT NO.; PROJECT NO.; TASK NO.; WORK UNIT NO.
11. TITLE (Include Security Classification): See Box 19	

12. PERSONAL AUTHOR(S): Robert I. Suizu, 1st Lt, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
MS Thesis	FROM _____ TO _____	March 1984	287

16. SUPPLEMENTARY NOTATION

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

17. COSATI CODES		
FIELD	GROUP	SUB. GR.
17	05	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Adaptive Estimation, Forward Looking Infrared (FLIR) sensor, Image Tracking, Kalman Filtering, Target Tracking, Multiple Model Adaptive Filter

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: ENHANCED TRACKING OF AIRBORNE TARGETS USING MULTIPLE MODEL FILTERING TECHNIQUES FOR ADAPTIVE FIELD OF VIEW EXPANSION

Thesis Chairman: Dr. Peter S. Maybeck

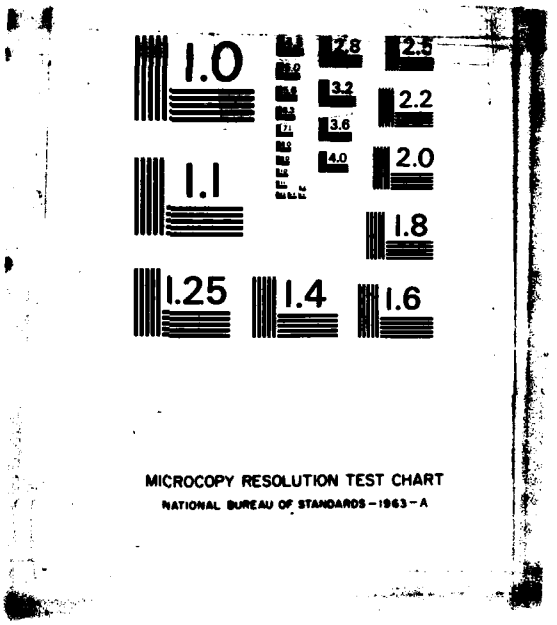
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: Dr. Peter S. Maybeck, Professor of Electrical Engineering	22b. TELEPHONE NUMBER (Include Area Code)

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE

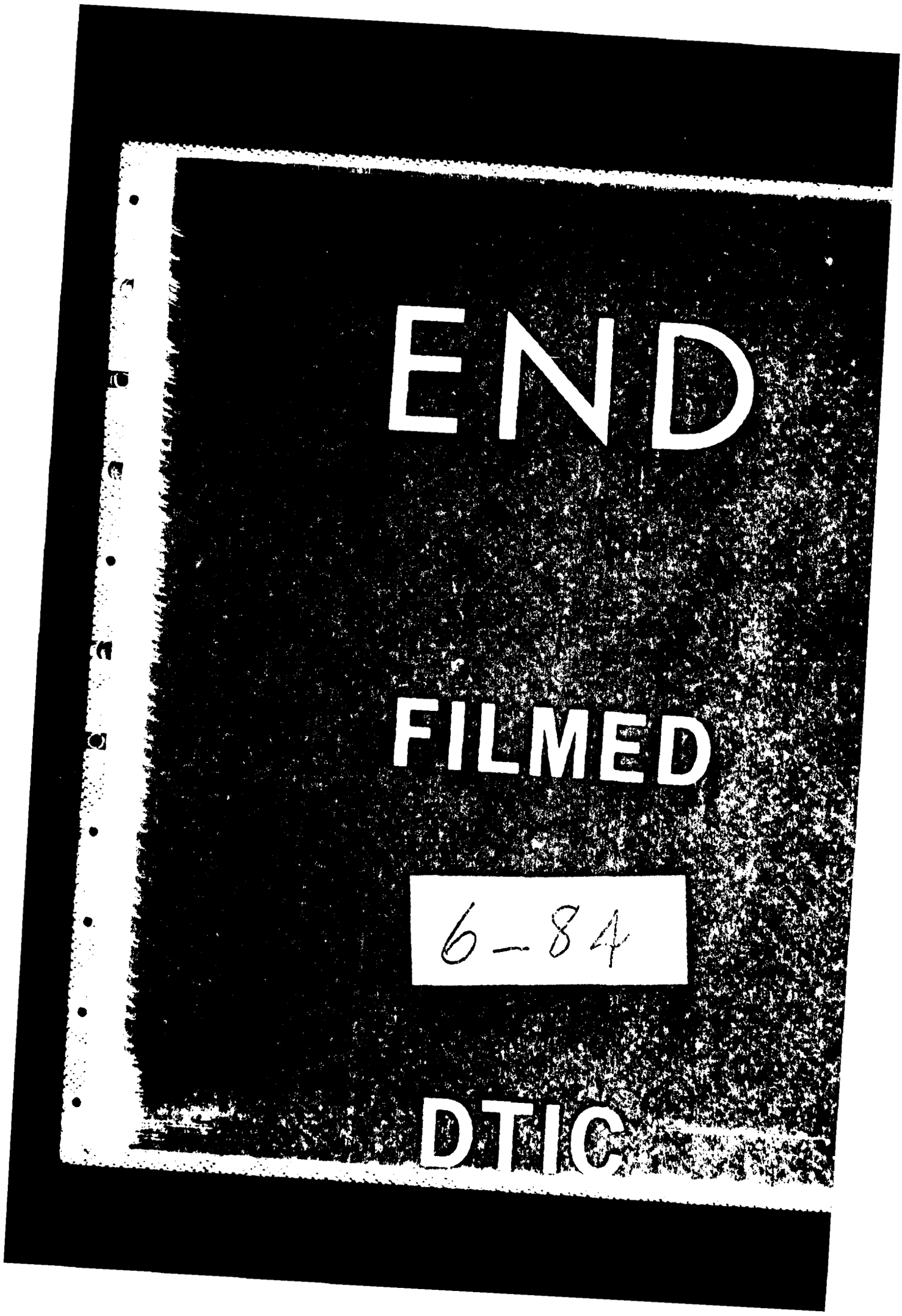
AD-A141 144 ENHANCED TRACKING OF AIRBORNE TARGETS USING MULTIPLE MODEL FILTERING TECH..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. R I SUIZU

4/4

UNCLASSIFIED DEC 83 AFIT/GE/EE/84M-4 F/G 12/1 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Previous studies at the Air Force Institute of Technology have developed two tracker algorithms which provide significant improvements in tracker performance against close-range, highly-dynamic, airborne targets, over a currently used direct correlation method. Digital signal processing techniques are used to derive a target shape function from available sensor information. In one formulation, this shape function is used in the measurement update portion of an extended Kalman filter to determin the target position offsets from the center of the sensor field of view. In the other tracker, the offsets are derived and incorporated into the tracking algorithm by using the shape function as a template for and enhanced correlator/linear Kalman filter structure. Combining these offsets with any a priori target information allows the tracker to produce better target position estimates than achievable from a conventional correlator. This research investigates using a multiple model approach for the adaptive expansion of the effective tracker field of view as a means of increasing the dynamic range of the tracker. Two independent Kalman filters, each receiving measurement information from a shared sensor, generate target position estimates. The multiple models are created by tuning the respective filters for "best" performance at differing condtions of exhibited target behavior and differing the physical size of their respective fields of view. Adaptive expansion of the tracker field of view is obtained by summing the weighted estimates of the two filters.


END
FILMED
6-84
DTIC